Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів і моделей побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості

(тема)

Виконав:
здобувач 2 року навчання,
групи ІУСТм-24-1
Олександр ТРЕБУНСЬКИХ
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник: доц. каф. ІУС Тетяна БОРИСЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ІУС _____ Костянтин ПЕТРОВ
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

" 24 " листопада 20 25 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Требунських Олександру В'ячеславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів і моделей побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості

затверджена наказом по університету від " 24 " листопада 2025 р. № 1055Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії " 17 " грудня 2025 р.

3. Вихідні дані до роботи матеріали передатестаційної практики, науково-технічні публікації та інтернет джерела з тематики кваліфікаційної роботи

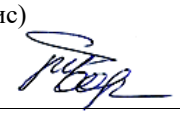
4. Перелік питань, що потрібно опрацювати в роботі аналіз методів та підходів для побудови рекомендацій житла в інформаційній системі агентства нерухомості; методів для побудови рекомендацій житла в інформаційній системі агентства нерухомості; розробка інформаційної технології побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості; реалізація модуля побудови рекомендацій; експериментальна перевірка наукових результатів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	24.11.2025	Виконано
2	Аналіз методів формування рекомендацій щодо вибору нерухомості	25.11.2025 – 27.11.2025	Виконано
3	Постановка задач дослідження	28.11.2025	Виконано
4	Дослідження та експериментальний вибір методу розрахунку ступеня подібності об'єктів	29.12.2025 – 01.12.2025	Виконано
5	Дослідження та експериментальний вибір методу перетворення значень текстових атрибутів об’єктів на вектори	02.12.2025 – 04.12.2025	Виконано
6	Розробка інформаційної технології побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості	05.12.2025 – 07.12.2025	Виконано
7	Розробка модуля побудови рекомендацій у інформаційній системі агентства нерухомості	08.12.2025 – 11.12.2025	Виконано
8	Оформлення пояснювальної записки до кваліфікаційної роботи	12.12.2025 – 13.12.2025	Виконано
9	Перевірка на плагіат	14.12.2025	Виконано
10	Попередній захист кваліфікаційної роботи	15.12.2025	Виконано
11	Захист роботи	18.12.2025	Виконано

Дата видачі завдання 24 листопада 2025 р.

Здобувач 
(підпис)

Керівник роботи 
(підпис) доц. каф. ІУС Тетяна БОРИСЕНКО
(посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 133 с., 14 рис., 20 табл., 2 дод., 24 джерела.

АГЕНТСТВО НЕРУХОМОСТІ, ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, МЕТОД, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, РЕКОМЕНДАЦІЇ, JAVA.

Об’єкт дослідження – процес формування рекомендацій для клієнтів у інформаційній системі агентства нерухомості.

Метою роботи є дослідження та експериментальний вибір методів та моделей для побудови рекомендацій користувачам у сфері нерухомості для підвищення якості та точності рекомендацій.

До використаних методів дослідження відносяться: зовнішня оцінка (Extrinsic Evaluation), методологія IDEF0, експериментальний підхід з використання синтетичних даних та методологія DFD.

Теоретичні результати – це опис аналізу та експериментального дослідження методів і моделей, які можна використовувати для формування рекомендацій у рамках content-based підходу, і розроблена інформаційна технологія побудови рекомендацій в інформаційній системі агентства нерухомості.

Практичні результати роботи представлені реалізованими компонентами веб-базованого модуля побудови рекомендацій.

Результати роботи можна використовувати при реалізації системи рекомендацій у специфіці ринку нерухомості та суміжних сферах.

Кваліфікаційну роботу виконано згідно методичних вказівок щодо розробки та оформлення кваліфікаційної роботи [1], ДСТУ 3008:2015 [2] і ДСТУ 8302:2015 [3].

ABSTRACT

Master's thesis: 133 pages, 14 figures, 20 tables, 2 appendices, 24 sources.

INFORMATION TECHNOLOGY, JAVA, METHOD, PROPERTY AGENCY, RECOMMENDATION SYSTEMS, RECOMMENDATIONS.

The object of research of the qualification work is the process of forming recommendations for clients in the information system of a property agency.

The purpose of the work is to research and experimentally select methods and models for building recommendations for users in the real estate sector in order to improve the quality and accuracy of recommendations.

The research methods used include: extrinsic evaluation, IDEF0 methodology, experimental approach using synthetic data, and DFD methodology.

Theoretical results – a description of the analysis and experimental study of methods and models that can be used to form recommendations within the framework of a content-based approach, and the developed information technology for building recommendations in the information system of a real estate agency.

Practical results are presented by the implemented components of the web-based recommendation building module.

The results of the work can be used in the implementation of a recommendation system in the real estate market and related areas.

The qualification work was performed in accordance with the methodological guidelines for the development and formatting of qualification work [1], DSTU 3008:2015 [2], and DSTU 8302:2015 [3].

ЗМІСТ

С.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

ІТ – інформаційна технологія

CLS – classification token

NLP – обробка природної мови

ВСТУП

З все більшим зростанням накопичення даних про клієнтів зростає попит на системи рекомендацій, які надають клієнтам персональні рекомендації. Великі компанії, що займаються електронною комерцією, все більше використовують системи рекомендацій, з метою зміцнення своєї конкурентної позиції. Дослідження систем рекомендацій постійно підтверджують пряму залежність між рівнем задоволеності споживачів та якістю рекомендацій, зокрема їх точністю і різноманіттям об’єктів, що рекомендує алгоритм [4].

Саме з цих обставин дослідження методів побудови рекомендацій є не лише доцільним, але й особливо актуальним у сучасних умовах тенденцій бізнесу, де здатність надавати персоналізовані пропозиції стає ключовим фактором успіху компанії на ринку.

Основною метою цієї роботи є дослідження методів розрахунку ступеня подібності об'єктів і методів та моделей перетворення текстів на вектори. Окрім цього, завданням цієї роботи є розробка інформаційної технології (ІТ) для побудови рекомендацій у інформаційній системі агентства нерухомості та реалізація програмного модуля на основі цієї технології.

1 АНАЛІЗ МЕТОДІВ ТА ПІДХОДІВ ДЛЯ ПОБУДОВИ РЕКОМЕНДАЦІЙ ЖИТЛА В СИСТЕМІ АГЕНТСТВА НЕРУХОМОСТІ

1.1 Формулювання проблеми побудови рекомендацій клієнтам щодо вибору нерухомості

Побудова рекомендацій – процес спроби передбачити, які об'єкти, наприклад: житло, фільми, музика, книги, новини тощо, будуть цікаві користувачеві, маючи певну інформацію про нього. Рекомендаційні технології здатні формувати персоналізовані рекомендації, ґрунтуючись на аналізі уподобань, поведінки та інтересів користувачів.

Системи рекомендацій є актуальною лінією захисту від проблеми надмірного вибору для користувачів. Через стрімке збільшення обсягу інформації, люди постійно зіштовхуються з величезною кількістю продуктів, фільмів або об’єктів житла, що ускладнює пошук справді відповідних для них пропозицій. Саме тому персоналізовані рекомендації стають ключовою стратегією для покращення користувацького досвіду. У цілому такі системи виконують суттєву та незамінну функцію у різних системах доступу до інформації, підтримуючи розвиток бізнесу, спрощуючи процес ухвалення рішень і широко застосовуючись у багатьох веб-сферах, включно з електронною комерцією та медійними платформами.

Зазвичай рекомендаційні списки створюються з урахуванням вподобань клієнтів, властивостей продуктів, історії взаємодій між ними, а також додаткової інформації, зокрема часових даних (наприклад, рекомендацій, що враховують послідовність дій) та просторових даних (як у випадку з підбором житла чи закладів) [5].

Використання модуля рекомендацій у складі інформаційної системи агентства нерухомості, що спеціалізується на довгостроковій оренді, може значно покращити користувацький досвід та привернути увагу втрачених користувачів. Завдяки аналізу поведінки користувачів, їхніх попередніх

пошуків, переглядів і обраних параметрів, цей модуль здатний пропонувати саме ті об’єкти житла, які найбільше відповідають індивідуальним потребам та інтересам конкретного клієнта.

У рамках цієї роботи має бути розроблений модуль рекомендацій для інформаційної системи агентства нерухомості, що спеціалізується на довгостроковій оренді, який буде надсилати електроні листи з рекомендаціями житла користувачам системи. Спрощене представлення роботи модуля можна побачити на діаграмі активності нижче (рис 1.1).

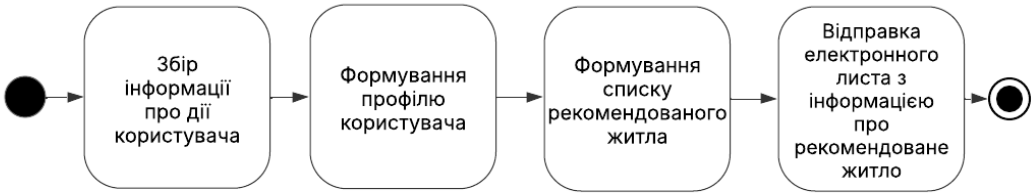


Рисунок 1.1 – Діаграма активності з спрощеним представленням роботи модуля

Під час використання системи зареєстрованим користувачем деякі його дії будуть фіксуватися, наприклад, перегляд сторінки житла або задання фільтру під час пошуку, після цього з цієї інформації буде формуватися профіль користувача. Профіль користувача – це векторне представлення уподобань користувача, сформоване на основі його дій у системі, що використовується для прогнозування інтересу до нових об’єктів.

Далі серед житла, яке не було переглянуто користувачем, буде обрано те яке найбільш підходить до профілю користувача на основі подібності. Після чого буде формуватися електронний лист, який відправляється користувачу.

Електронні листи з рекомендаціями будуть надсилатися користувачу з періодичністю, зазначеною у таблиці 1.1.

Таблиця 1.1 – Періодичність надсилання електронних листів з рекомендаціями користувачам

Остання активність (t)	Частота надсилання
$t \leq 7$ днів	Раз на чотири дні (наприклад, можна відправляти листи у 1, 5, 9, 13, 17, 21, 25, 29 числах)
7 днів $< t \leq$ 1 місяць	Раз на 8 днів(наприклад у 1, 9, 17, 25 числах)
1 місяць $< t \leq$ 5 місяців	Раз на місяць (наприклад, кожного 1 числа)
$t >$ 5 місяців	Раз у чотири місяці

Окрім цього у листах рекомендацій не має бути житла яке користувач вже переглядав та яке йому вже рекомендували за останні п’ять місяців. Кожен лист має містити інформацію (фото, пріоритетні дані та посилання) про шість рекомендованих об’єктів житла, якщо користувач цікавився різними типами житла, наприклад, квартири та приватні будинки. Рекомендаційний лист має містити різні типи житла у пропорції, відповідній до інтересу користувача.

Модуль має ураховувати під час формування рекомендацій такі параметри: орендна плата, адреса (місто, район), кількість кімнат, площа, поверх, вільний опис, близькість до метро, ремонт, наявність побутової техніки (пральної машини, посудомийної машини, бойлера, обігрівача тощо), походження об'єкта та кількість спальних місць.

Модуль має коректо працювати у рамках наявної інформаційної системи агентства нерухомості.

Модуль має відсилати рекомендаційні листи з урахуванням дій користувача, з певною частотою, відповідно до активності користувача, з урахуванням нещодавніх рекомендацій, має ураховувати основні параметри житла та коректо працювати у рамках наявної інформаційної системи агентства нерухомості.

1.2 Аналіз методів формування рекомендацій щодо вибору нерухомості

З урахуванням сучасних тенденцій розширення використання персональних рекомендацій у різних сферах, вони є важливим елементом будь якої системи, яка направлена на роботу з клієнтами та має широкий асортимент.

Існують декілька основних підходів до створення персональних рекомендацій.

Колаборативна фільтрація – це підхід до створення рекомендацій, що передбачає використання наявних оцінок від групи користувачів для прогнозування невідомих уподобань іншого користувача. Його головна ідея полягає в тому, що люди, які раніше подібно оцінювали однакові об’єкти, зазвичай мають схожі інтереси й надалі, імовірно, надаватимуть подібні оцінки іншим елементам [6].

Наприклад, за допомогою колаборативної фільтрації музичний сервіс здатен визначити, які треки можуть зацікавити користувача, навіть якщо відома лише частина його вподобань чи антипатій. Система створює персоналізовані рекомендації для кожного, спираючись на сукупні дані, отримані від великої кількості користувачів.

До невирішених проблем колаборативної фільтрації відносять:

а) розрідженість даних – переважна частина користувачів не залишає оцінок для більшості з товарів, тому формується велика, але розріджена матриця «предмет – користувач», що ускладнює обчислення рекомендацій;

б) масштабування – із розширенням бази користувачів у системі виникає проблема ефективного масштабування: обчислення, необхідні для роботи алгоритмів колаборативної фільтрації, стають надто складними та потребують значних ресурсів;

в) проблема холодного старту – поява нових користувачів чи нових об’єктів суттєво ускладнює роботу рекомендаційних систем, що ґрунтуються

на цьому метод;

г) різноманітність — спочатку колаборативну фільтрацію розробляли з метою підвищення різноманітності рекомендацій, щоб користувачі мали змогу відкривати нові товари серед великого вибору. Однак на практиці деякі алгоритми, особливо ті, що спираються на показники продажів чи рейтинги, часто створюють несприятливі умови для просування нових або маловідомих продуктів;

ґ) наявність «білих ворон» – це проблема пов’язана з користувачами, чиї вподобання суттєво відрізняються від більшості, що створює труднощі для систем колаборативної фільтрації.

Content-based підхід – це підхід, при якому використовуються властивості об’єкта, щоб запропонувати інші об’єкти, подібні до тих, які вже сподобалися користувачеві, ґрунтуючись на його попередній активності або явних оцінках.

До проблем цього підходу відносять:

а) розрідженість даних та проблему холодного старту, які вже були описані у контексті колаборативної фільтрації вище, з єдиною різницею, що content-based немає проблем холодного старту при додаванні нових товарів;

б) ефект «бульбашки фільтрації» – користувач отримує в основному те, що схоже на попередні уподобання, що обмежує знайомство з чимось новим та може призвести до набридання та втратити інтересу;

в) важливість повноти характеристик об'єктів – підхід сильно залежить від того, як описані об'єкти. Якщо характеристики бідні, неповні або погано відображають те, що дійсно подобається користувачеві рекомендації будуть слабкими.

Демографічний підхід (Demographic Filtering) – це підхід, що використовує демографічні дані(стать, вік, освіта) користувача для визначення, які товари можуть бути підходящими для рекомендації. Він не має проблеми появи нових користувачів, оскільки не спирається на рейтинги під час формування рекомендацій. Водночас підхід потребує збирання

достатнього обсягу демографічної інформації, що звужує можливості його застосування. Часто його поєднують з іншими рекомендаційними методами як додатковий інструмент для підвищення якості результатів [7, 8].

До мінусів демографічного підходу можна віднести:

а) конфіденційність – використання демографічних даних стикається з певними проблемами з точки зору конфіденційності та закону;

б) точність – знижена точність порівняно з content-based або іншими популярними підходами;

в) необхідність демографічних даних – система с самого початку має мати достатню кількість демографічних даних для роботи.

Підхід на основі знань (Knowledge-Based Filtering) – Це особливий підхід, який базується на явних знаннях про асортимент товарів, уподобання користувачів та критерії рекомендацій, а саме на тому, який товар варто рекомендувати в певному контексті. Цей підхід використовується в ситуаціях, коли альтернативні підходи, зокрема колаборативний або content-based, неможливо застосувати.

Однією з ключових переваг рекомендаційних систем на основі знань виступає те, що вони не стикаються з проблемою холодного старту. Проте їхнім суттєвим недоліком стає потенційне обмеження під час отримання знань, що пов'язано з потребою чітко формулювати рекомендаційні знання.

Такий підхід демонструє високу ефективність у складних сферах із нечастими покупками товарів, адже рейтингові механізми там зазвичай працюють погано через брак достатньої кількості оцінок.

Гібридний підхід – підхід, що передбачає комбінування інших підходів до формування рекомендацій з метою використання їхніх взаємодоповнюючих сильних сторін [8].

До головних проблем гібридного підходу відносять такі:

а) складність реалізації та підтримки – гібридний підхід включає комбінування різних технік, що вимагає більше ресурсів, часу на розробку та обслуговування;

б) обчислювальні витрати – комбінування різних підходів означає більше обчислень, що може загострити проблеми з продуктивністю та часом відгуку.

Як можна побачити зі списку зверху, є достатньо підходів для створення рекомендацій, але у цій роботі вважаю доцільним розглянути content-based підхід через такі його переваги:

а) максимальна персоналізація рекомендацій – рекомендації базуються виключно на даних про конкретного користувача, а не на поведінці інших людей, це дозволяє враховувати індивідуальні смаки та особливості користувача;

б) відсутність залежності від інших користувачів – на відміну від колаборативних фільтрів, не потрібна велика база користувачів зі схожими інтересами. Працює навіть при невеликій аудиторії, бо достатньо даних про самого користувача та дані контенту який він переглядав;

в) чіткість – у цьому підході легко пояснити, чому була зроблена рекомендація, це спрощує налагодження і підвищує контроль над роботою системи;

г) легка робота з новими об’єктами – нові об’єкти можна відразу рекомендувати, якщо є їх опис, немає проблеми холодного старту для об'єктів;

ґ) простота реалізації – для тестової версії достатньо простих алгоритмів обчислення схожості між векторами ознак, легко масштабується і комбінується з іншими методами.

При використанні content-based підходу формують профіль користувача, на основі відповідності до нього відбираються найбільш підходячи, для клієнта, об’єкти. Для виявлення відповідності житла існують різні методи для порівняння векторів об’єктів та роботи з вільним текстом.

Розглянемо методи, які можна використовувати для порівняння векторів об’єктів.

Косинусна подібність (Cosine Similarity) – це метод, необхідний для порівняння векторів об'єктів, що є мірою подібності двох ненульових векторів.

Цей метод визначає напрямок векторів, а не їх величину. Це пояснюється тим, що вона обчислює кут між векторами і використовує його як міру подібності. Ця міра є однією з найпоширеніших мір подібності [9].

Евклідова відстань – метод, який представляє собою метрику в евклідовому просторі, що визначає відстань між парою точок евклідового простору шляхом обчислення згідно з теоремою Піфагора [10].

Евклідова відстань представляє собою елементарний спосіб визначення відстані між об'єктами. Цей метод застосовується у різноманітних сферах для розв'язання задач, що стосуються простору та відстані. На противагу косинусній подібності, котра бере до уваги кут між двома точками чи векторами, евклідова відстань зосереджується на довжині між ними.

Відстань Гауера – метод, що представляє собою міру подібності, котра визначає різницю між векторами з комбінуванням числових та категоріальних атрибутів. На противагу традиційним метрикам відстані, зокрема евклідовій відстані, яка орієнтована на обробку виключно числових даних, відстань Гауера ефективно працює з наборами даних, що включають обидва типи атрибутів. Цей метод обчислює матрицю відстаней, яка кількісно визначає ступінь відмінності між двома точками даних, при цьому менша відстань свідчить про більшу подібність між точками.

Відстань Гауера комбінує різноманітні метрики відстані для кожного типу атрибуту та визначає загальну відстань між двома точками як середнє арифметичне індивідуальних відстаней атрибутів. Відстань Гауера є ефективним методом для визначення відмінностей між окремими елементами у наборах даних зі змішаними типами даних.

Манхеттенська відстань (Manhattan distance or taxicab metric) – метод, що є сумою зважених довжин відрізків, які сполучають точки, кожен з яких паралельна координатній осі [11].

Відстань Хеммінга (Hamming distance) – це метод, що визначає число позицій, у яких відповідні значення двох векторів ідентичної довжини є різними. Згідно з визначенням цієї метрики, вона не бере до уваги конкретні

значення векторів, окрім факту їх відмінності чи рівності.

Відстань Махаланобіса (Mahalanobis distance) – метод, що представляє собою метрику, котра використовує концепцію евклідової відстані.

Відстані Махаланобіса ґрунтуються на положенні та дисперсії багатовимірного нормального розподілу та дає змогу визначити, наскільки віддалена кожна точка простору від центральної частини цього розподілу [12].

По своїй суті вона є подібною до евклідової відстані, проте містить додатковий зміст, чим ближче розташована точка до «центру мас», тим вища ймовірність того, що це саме та точка, яка нам потрібна.

Особливості методів розрахунку ступеня подібності об'єктів представлені у таблиці 1.2.

Таблиця 1.2 – Особливості методів розрахунку ступеня подібності об'єктів

Назва	Особливості
Косинусна подібність (Cosine Similarity)	Вимірює подібність векторів без врахування їх довжини, добре працює з розрідженими векторами.
Евклідова відстань (Euclidean distance)	Простий, інтуїтивний, фокусується на відстані між точками.
Відстань Гауера (Gower Distance)	Підходить для змішаних типів даних
Манхеттенська відстань (Manhattan distance)	Фокусується на змінних відстанях вздовж осей координат
Відстань Хеммінга (Hamming distance)	Видає кількість позицій, в яких відповідні значення двох векторів однакової довжини відрізняються
Відстань Махаланобіса (Mahalanobis distance)	Працює відповідно Евклідової відстані, але враховує відстань до «центру мас»

Методи та моделі, які можна використовувати для перетворення текстових характеристик об’єктів на вектори, інакше кажучи для ембедінгу, описуються далі. Вони потрібні для роботи з вільним текстом.

Метод Bag-of-Words – цей метод призначений для спрощеного відображення тексту, що застосовується в обробці природних мов та

інформаційному пошуку. У даній моделі текст відображається у формі набору його слів без урахування граматичних правил та послідовності слів, однак зі збереженням інформації про їхню кількість.

TF-IDF – цей метод представляє собою вдосконалену версію методу Bag-of-Words, котрий не лише підраховує частоту появи слова, а також враховує ступінь важливості цього слова для конкретного тексту відносно всього набору документів.

За визначенням, TF IDF є метрикою, що обчислюється як добуток двох величин TF і IDF. Де TF позначає частоту вживання терміна, а IDF обернену частоту документа [13].

Word2Vec – це метод ефективно оброблює природну мову для отримання векторних представлень слів. Ці вектори фіксують інформацію про значення слова з урахуванням сусідніх слів. Крім роботи зі словами, деякі його концепції виявилися ефективними в розробці рекомендаційних механізмів і наданні сенсу даним навіть у комерційних, немовних завданнях. Цю технологію застосували у своїх двигунах рекомендацій такі компанії, як Airbnb, Alibaba, Spotify і Anghami.

BERT (Bidirectional Encoder Representations from Transformers або двоспрямовані кодувальні представлення з трансформерів) – універсальна мовна модель, розроблена для попереднього навчання глибоких двонаправлених представлень на немаркованому тексті через одночасне врахування як лівого, так і правого контексту на всіх шарах. Після попереднього навчання BERT-подібну модель можна точно налаштувати, додавши лише один вихідний шар для створення найсучасніших моделей широкого спектру задач від векторизації тексту до мовного виводу без значних архітектурних модифікацій під конкретні завдання. BERT є концептуально простою та емпірично потужною моделлю. Вона показала найкращі результати в одинадцяти задачах обробки природної мови, включаючи підвищення бала GLUE до 80,5% (абсолютне покращення 7,7%), точність MultiNLI до 86,7% (абсолютне покращення 4,6%), F1 для SQuAD v1.1 до 93,2

(абсолютне покращення 1,5 пункту), F1 для SQuAD v2.0 до 83,1 (абсолютне покращення 5,1 пункту) [14].

Особливості методів для виконання ембедінгу вільного тексту представлені у таблиці 1.3.

Таблиця 1.3 – Особливості методів для перетворення текстових атрибутів об’єктів на вектори

Назва	Особливості
Метод Bag-of-Words	Ігнорує порядок слів, але враховує їх кількість
Метод TF-IDF	Працює як Bag-of-Words, але зменшує важливість часто вживаних слів
BERT-подібна модель	Ефективно оброблює природну мову з використанням нейронної мережи

1.3 Обґрунтування вибору методів та моделей формування рекомендацій для експериментального порівняння

У рамках цієї роботи для експериментального порівняння та вибору будуть розглянуті такі найбільш поширені методи для порівняння векторів об’єктів (розрахунку ступеня подібності об’єктів):

а) косинусна подібність;

б) евклідова відстань;

в) відстань Гауера.

Косинусна подібність була обрана через такі переваги:

а) простота обчислення – формула дуже проста, швидко реалізується та добре оптимізується;

б) добре працює з розрідженими даними – коректно працює з розрідженими даними, не спотворюючи результат через нулі.

До переваг, через які була обрана евклідова відстань, відносяться такі:

а) проста інтерпретація – вимірюється фактична «пряма» відстань між двома точками в просторі, що має чітке геометричне трактування, легко зрозуміти та пояснити без глибоких математичних знань;

б) легка у реалізації – проста формула, реалізується у будь-якій мові програмування без великої кількості коду;

в) ефективна для числових даних – добре працює, коли всі ознаки мають однакову природу, одиниці вимірювання та масштаб.

Відстань Гауера має такі переваги:

а) передбачена підтримка змішаних типів ознак – може одночасно працювати з числовими, категоріальними та бінарними даними;

б) масштабування кожної ознаки – кожна ознака нормується окремо у діапазон [0,1], тому всі вони роблять співставний внесок у результат. Це вирішує проблему різних одиниць вимірювання;

в) стійкість до пропущених значень – коректно обробляє розріджені дані, якщо значення відсутні, ця ознака просто не враховується у розрахунку між конкретною парою об’єктів;

г) інтерпретованість – значення відстані завжди у межах [0,1], де 0 означає, що об’єкти ідентичні, а 1, що об’єкти максимально різні. Це спрощує аналіз і порівняння результатів;

ґ) гнучкість у вагуванні ознак – можна задавати ваги для кожної ознаки, що дозволяє підкреслити важливі характеристики й зменшити вплив другорядних.

Для експериментального порівняння та вибору методу перетворення текстових атрибутів об’єктів на вектори (виконання ембедингу) в роботі будуть розглянуті такі методи та модель:

а) метод Bag-of-Words;

б) метод TF-IDF;

в) модель нейронної мережі з архітектурою BERT .

Метод Bag-of-Words був обраний через такі переваги:

а) простота – легко розраховується та реалізується;

б) легковесність – не потребує великих обчислювальних ресурсів порівняно з сучасними мовними моделями;

в) ефективність у простих задачах – для коротких текстів часто дає достатньо точні результати, особливо якщо тексти мають характерні ключові слова.

Метод TF-IDF забезпечує такі ключові переваги:

а) простота – легко розраховується та реалізується;

б) легковесність – не потребує великих обчислювальних ресурсів порівняно з сучасними мовними моделями;

в) захищеність від часто вживаних слів – зменшує вагу часто вживаних слів, наприклад: «та», «це» або «на», і підсилює рідкісні, але змістовні терміни, це дозволяє ефективно визначати ключові слова в тексті;

г) немає потреби в навчанні – на відміну від нейромереж або мовних моделей, TF-IDF не вимагає тривалого чи складного процесу тренування, оскільки його робота базується на статистичному аналізі частоти слів у документах, а не на навчанні з великих обсягів даних.

До сильних сторін моделей нейронної мережі з архітектурою BERT можна віднести:

а) контекстуальне розуміння, на відміну від Bag-of-Words та TF-IDF, BERT аналізує слова з урахуванням їхнього контексту в реченні;

б) висока точність на складних задачах – BERT демонструє значно кращі результати в задачах класифікації текстів, аналізу тональності, семантичної подібності та пошуку відповідей на питання, особливо коли тексти містять складні граматичні конструкції або багатозначні слова.

1.4 Постановка задач дослідження

Головною задачею роботи є дослідження методів та моделей,

необхідних для реалізації процесу побудови рекомендацій щодо об’єктів нерухомості клієнтам в інформаційній системі агентства нерухомості, а саме методів для порівняння об’єктів житла та векторизації тексту.

Об’єктом дослідження магістерської роботи є процес побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості.

Предметом дослідження є методи та моделі, що можна використовувати для побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості.

Проблемою дослідження є необхідність облегшення вибору для користувача прі надмірній кількості об’єктів, користувачі часто стикаються з незліченною кількістю пропозицій, що ускладнює підбір кращої з них.

Метою дослідження є виявлення найбільш підходящих методів для побудови рекомендацій та використання їх у інформаційній системі агентства нерухомості для розсилання рекомендаційних листів на пошту зареєстрованим користувачам.

При виконанні даної роботи мають бути вирішені такі задачі дослідження:

а) аналіз існуючих методів розрахунку ступеня подібності об’єктів та перетворення текстових атрибутів об’єктів на вектори;

б) дослідження та експериментальний вибір методу розрахунку ступеня подібності об’єктів (косинусної подібності, евклідової відстані, відстані Гауера) для використання при побудові рекомендацій клієнтам в інформаційній системі (ІС) агентства нерухомості;

в) дослідження та експериментальний вибір методу або моделі для перетворення текстових атрибутів об’єктів на вектори (Bag-of-Words, TF-IDF, BERT) для використання при побудові рекомендацій клієнтам в ІС агентства нерухомості;

г) розробити інформаційну технологію побудови рекомендацій клієнтам в ІС агентства нерухомості;

ґ) виконати програмну реалізацію інформаційної технології побудови

рекомендацій клієнтам в ІС агентства нерухомості;

д) виконати експериментальну перевірку отриманих наукових результатів.

2 ДОСЛІДЖЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ ВИБІР МЕТОДІВ ДЛЯ ПОБУДОВИ РЕКОМЕНДАЦІЙ ЖИТЛА В ІНФОРМАЦІЙНІЙ СИСТЕМІ АГЕНТСТВА НЕРУХОМОСТІ

2.1 Вибір методу розрахунку ступеня подібності об'єктів

У роботі будуть досліджені три методи розрахунку ступеня подібності об’єктів: косинусна подібність, евклідова відстань та відстань Гауера. Далі описаний підхід до вибору кращого з них.

2.1.1 Підхід до вибору кращого методу розрахунку ступеня подібності об'єктів

Оцінка методів розрахунку ступеня подібності об'єктів у контексті рекомендаційних систем є доволі складною задачею, і не існує універсальний засіб до її вирішення. Основні методи перевірки діляться на два наступні підходи:

а) офлайн-оцінка (Offline Evaluation) – підхід оцінювання задоволеності користувачів рекомендаціями, використовуючи статичні дані з попередніх взаємодій користувачів. Ці оцінки надають приблизні оцінки ефективності системи. Однак офлайн-оцінка не може точно оцінити нові рекомендації, оскільки вони відсутні в історичних даних і не можуть бути оцінені як релевантні [15]. До офлайн-оцінок відносять:

1) Precision@K – метрика якості, яка визначає точність як частку релевантних елементів у топ-k рекомендацій [16];

2) Recall@K – метрика, яка показує, яку частку всіх релевантних об'єктів модель знайшла у своїх перших K результатах;

3) Mean Average Precision (MAP) – метрика для оцінки якості

ранжування, яка враховує точність на кожному знайденому релевантному об'єкті та потім усереднює результат;

б) онлайн-оцінка (Online Evaluation) – підхід при якому алгоритми рекомендацій впроваджується в онлайн-середовище з метою порівняння їхньої продуктивності [17]. До методів онлайн-оцінювання відносять:

1) A/B-тестування – форма перевірки гіпотез, при якій два варіанти програмного забезпечення порівнюються в реальних умовах з точки зору кінцевого користувача [18]. При цьому підході новий алгоритм тестується на підгрупі користувачів і порівнюється з результатами контрольної групи, яка тестувала старий алгоритм. Це найкращий спосіб порівняти новий алгоритм зі старим;

2) тестування методом «багаторукого бандита» – це складний підхід до оцінки алгоритмів, який динамічно розподіляє трафік між різними алгоритмами залежно від їх ефективності. Цей метод дозволяє збалансувати необхідність дослідження і експлуатації, тобто перевіряє різні алгоритми, але віддає перевагу найбільш ефективним моделям;

3) чергування – метод при якому рекомендації різних алгоритмів надаються одним і тим же користувачам одночасно. Це досягається шляхом чергування елементів з різних алгоритмів у списку рекомендацій. Відстежуючи взаємодію користувачів з цими чергуючимися списками.

В умовах відсутності можливості нормального використання онлайн-оцінки або використання історичних даних, вважаю доцільним взяти підхід з використання синтетичних або підготовлених датасетів, цей підхід має наступні переваги:

а) не потрібні реальні дані про поведінку користувачів;

б) повний контроль над умовами тестування;

в) не потрібно чекати реакції реальних користувачів, як при A/B тестуванні;

г) можна тестувати до публічного запуску;

ґ) можна створити рідкісні або екстремальні сценарії.

У рамках цього підходу для перевірки методів були використані синтетичні вхідні дані, які складалися із дванадцяти житлових об'єктів. Ці об'єкти були розподілені на три категорії («Перша», «Друга» та «Третя»). У кожну категорію було включено по п'ять об'єктів – три з них виступали як основні представники категорії (вони є схожими один на одного), а два як такі, що частково поєднують окремі значення атрибутів двох різних категорій.

Візуальне відображення підготовлених даних можна побачити на рисунку 2.1.

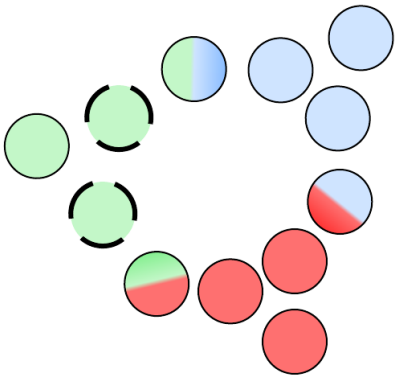


Рисунок 2.1 – Візуальне відображення підготовлених для експериментів вхідних даних

На цьому рисунку об’єкти розділені категоріями відмічені різними кольорами, а часткові представники відразу двома кольорами категорій до яких вони частково відносяться, пунктиром відмічені об’єкти, які переглянуті користувачем.

При оцінюванні методів будуть враховуватися критерії, які представлені у таблиці 2.1.

Таблиця 2.1 – Критерії оцінювання методів розрахунку ступеня подібності об'єктів

Критерій оцінювання	Опис критерія
Позиції у рейтингу об'єктів	Це легкий та інтуїтивний спосіб виявити відповідність результатів
Дискримінативна здатність (у %)	Дозволяє виявити, наскільки метод відрізняє однокатегоріальні об’єкти від інших
Підтримка категоріальних значень	Важлива, бо у сфері застосування є багато категоріальних значень
Вразливість до екстра значень	Впливає на коректність результатів у незвичайних умовах
Потреба в налаштуванні коефіцієнтів важливості атрибутів	Коефіцієнти важливості необхідні через різний ступінь впливу різних атрибутів об'єкту на вибір

Формула розрахунку дискримінативної здатності наведена нижче:

$$d = \frac{a}{(b/100)}$$

де d – дискримінативна здатність;

a – середнє значення подібності непереглянутих об’єктів житла категорії «Перша»;

b – середнє значення подібності усіх непереглянутих об’єктів житла.

Для повноцінного відбору оптимального методу розрахунку ступеня подібності об'єктів в межах предметної області необхідно визначити метод для роботи з вільним текстом. Тому на основі первинного аналізу, проведеного у розділі 1.2, як попередньо обрана була взята реалізація нейронної моделі sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2, яка є моделлю класу Sentence-BERT (архітектурна модіфікація BERT). Тестові дані житла, які використовувались при перевірці методів для роботи з числовими та категоріальними даними, наведені на рисунку 2.2.

Переглянутими були відмічені два об’єкти житла з категорії «Перша» – вони в стовпці «visited» мають значення 1, а не NULL. Інші об’єкти категорії «Перша» у рамках перевірки вважаються релевантними до переглянутих.

category	visited	property_id	city	col_rooms	description	col_beds	price	area	district	floor	type	equipments
Перша	1	1	Харків	2	В квартирі 2 кімнати, Технологія будівництв...	2	6000	42	Салтівський	4	квартира	З ремонтом, Кондиціонер, Кухонна плита, Ліфт, Мікрохвильова піч, Можна з деякими тв...
Перша	1	2	Харків	2	Здам в оренду 2 кімнатну квартиру на комф...	2	5000	43	Салтівський	1	квартира	Зі старим ремонтом, Кондиціонер, Кухонна плита, Ліфт, Можна з деякими тваринами, Ро...
Перша	NULL	3	Харків	3	Довготривала оренда. Ювілейний, 59 (пору...	3	5000	33	Салтівський	3	квартира	З ремонтом, Кухонна плита, Ліфт, Можна з деякими тваринами, Можна з дітьми
Перша-друга	NULL	4	Харків	1	Сдам 1 комнатную квартиру. В квартире ест...	2	4000	17	Індустріальний	5	квартира	Вантажний ліфт, Зі старим ремонтом, Конс'єрж, Кухонна плита, Ліфт, Роздільний санвузол
Перша-третя	NULL	5	Харків	3	Генераторы ! Низкий этаж ! Аренда 3к Lux А...	5	14000	54	Основянський	5	квартира	З ремонтом, Кондиціонер, Кухонна плита, Ліфт, Мікрохвильова піч, Пральна машина
Друга	NULL	6	Харків	1	Здам свою кімнату в блоці з 6 кімнат. Кімнат...	2	2000	13	Індустріальний	1	кімната	З ремонтом, Мікрохвильова піч, Можна з деякими тваринами
Друга	NULL	7	Харків	1	Сдам кімнату район ХТЗ. 4 поверх п'яти пове...	2	1700	10	Індустріальний	4	кімната	Зі старим ремонтом, Пральна машина
Друга	NULL	8	Харків	1	Сдам 1 комнату в частном секторе. 8 кв.м. Н...	1	3000	8	Індустріальний	1	кімната	З ремонтом, Кондиціонер, Кухонна плита, Можна з деякими тваринами, Пральна машин...
Друга-третя	NULL	9	Харків	1	Здаються затишні гостінки (номери) за адре...	2	15000	70	Основянський	1	кімната	NULL
Третя	NULL	10	Харків	4	сдам дом со всеми удобствами!стиралка пыл...	10	18000	100	Основянський	NULL	приватний будинок	З ремонтом, Мікрохвильова піч, Пральна машина, Роздільний санвузол
Третя	NULL	11	Харків	4	Сдам дом, 2х этажный, 2 х комнатный , с об...	5	16800	58	Основянський	NULL	приватний будинок	З ремонтом, Кондиціонер, Мікрохвильова піч, час до метро 6-10 хв
Третя	NULL	12	Київ	5	Сдается в долгосрочную аренду домик (1ко...	5	17000	55	Основянський	NULL	приватний будинок	З ремонтом, Кухонна плита, Мікрохвильова піч, Пральна машина, час до метро 10-20 хв

Рисунок 2.2 – Тестові дані об’єктів житла

Тестові дані двох фільтрів, які використовувались при перевірці, наведені на рисунку 2.3.

filter_id	types	max_price	col_rooms	col_beds	districts	city	equipments
1	квартира	14000	1	3	Салтівський, Київський	Харків	З ремонтом, Зі старим ремонтом, Кухонна плита, Мікрохвильова піч, Можна з деякими тваринами, Роздільний санвузол
2	квартира	7000	3	2		Харків	З ремонтом, Кондеціонер, Кухонна плита, Мікрохвильова піч, Можна з деякими тваринами, Пральна машина

Рисунок 2.3 – Тестові дані фільтрів

Черговість дій щодо перевірки методів розрахунку ступеня подібності об'єктів житла наведено на рисунку 2.4.

Рисунок 2.4 – Етапи перевірки методів розрахунку ступеня подібності об'єктів житла

На першому кроці створюється узагальнений профіль користувача з використанням переглянутих об'єктів житла і використаних фільтрів. Потім для кожного непереглянутого об’єкта житла розраховується ступінь схожості

із профілем користувача. Далі для об'єктів, які не відповідають фільтрам, що використовував користувач, знижується оцінка, в залежності від рівня порушення фільтрів, визначеного за методом, що перевіряється. Після чого будується рейтинг непереглянутих об'єктів на основі ступеню подібності тільки профілю користувача.

У ході оцінювання об’єктів житла на відповідність профілю бралися до уваги такі атрибути житла, які увійшли до профіля. Їх перелік наведений в табл. 2.2.

Таблиця 2.2 –Атрибути профіля користувача

Назва атрибуту	Тип даних атрибуту	Джерело значення атрибуту
Перелік населених пунктів	Колекція унікальних текстових значень	Переглянуте житло
Діапазон кількості кімнат у житлі	Об’єкт зі статистикою	
Описи житла, у виді векторів, об’єктів житла	Список векторів	
Діапазон кількості спальних місць у житлі	Об’єкт зі статистикою	
Діапазон орендних плат	Об’єкт зі статистикою	
Діапазон площ житла	Об’єкт зі статистикою	
Перелік районів	Колекція унікальних текстових значень	
Діапазон поверхів	Об’єкт зі статистикою	
Перелік типів об’єктів житла	Колекція унікальних текстових значень	
Перелік приладів та додаткових параметрів об’єктів житла та їх кількість серед переглянутих об’єктів	Колекція пар ключ–значення	
Перелік типів житла	Колекція унікальних текстових значень	Задані фільтри

Продовження таблиці 2.2

Назва атрибуту	Тип даних атрибуту	Джерело значення атрибуту
Максимальна ціна	Числовий тип	Задані фільтри
Перелік кількостей кімнат	Колекція унікальних числових значень	
Назва складової профіля	Тип даних складової профіля	
Перелік кількостей спальних місць	Колекція унікальних числових значень	
Перелік районів	Колекція унікальних текстових значень	
Перелік приладів та додаткових параметрів	Колекція пар ключ–значення	
Перелік населених пунктів	Колекція унікальних текстових значень	

У ході корекції (зменшення) оцінки за порушення умов фільтрів враховувалися такі параметри: назва населеного пункту, район житла, кількість спальних місць, кількість кімнат, максимальна орендна плата, тип житла, обладнання та побічні параметри.

2.1.2 Дослідження та перевірка методу косинусної подібності

У методі косинусної подібністі розраховується міра подібності між двома векторами передгілбертового простору, яка використовується для вимірювання косинуса кута між ними. У разі порівняння двох об'єктів, косинусна подібність двох об'єктів змінюється в діапазоні від 0 до 1, де 1 повна збіжність, а 0 абсолютна незбіжність.

Формула косинусної подібності виглядає наступним чином:

$$cosine\_similarity\ (A,B)\ =\ \frac{A*B}{||A||*||B||}, \tag{2.1}$$

де A*B – скалярний добуток векторів;

||A|| – норма вектору A;

||B|| – норма вектору B [19].

Одна з причин популярності косинусної подібності полягає в тому, що вона ефективна як оціночний показник, особливо для розріджених векторів, оскільки необхідно враховувати тільки ненульові вимірювання.

Рекомендаційні системи та великі мовні моделі застосовують косинусну подібність для визначення найбільш релевантного контенту та вибору відповідей, що мають найвищу семантичну значущість.

В обох випадках косинусна подібність оцінює рівень відповідності між отриманими векторами, сприяючи виявленню закономірностей і зв’язків у складних масивах даних. Вона відіграє ключову роль у галузі рекомендаційних систем використовуючись як алгоритми пошуку за подібністю, щоб пропонувати товари, медіа або контент, що узгоджуються з поведінкою та вподобаннями користувачів.

Далі описуються деталі реалізації для тестування методу косинусної подібності.

Оскільки в профілі користувача числові значення (кількість кімнат, спальних місць, орендна плата, площа, поверх) задані у вигляді діапазонів, то під час порівняння кожної координати вектору береться те значення з діапазону профілю, яке найближче до відповідного значення вектору житла.

Наприклад, якщо значення кількості кімнат у житлі 3, а у профіля діапазон кількості кімнат дорівнює 5-8, то при порівнянні будуть взяті числа 3 та 5.

Для категоріальних значень (місто, район, тип житла) використовується такий підхід: якщо значення житла збігається з одним із значень у списку профілю, то при порівнянні і житло, і профіль отримують значення 1. Якщо ж збігу немає житло отримує значення 0, а профіль 1. В залежності від

важливості атрибута можуть додаватися різні коефіцієнти Використання цього підходу для значення атрибуту «район», можна побачити у лістингу 2.1.

Лістинг 2.1 – Використання підходу розрахунку подібності до категоріальних значень, для значення атрибуту «район» (фрагмент файлу CosineSimilarity.java)

```
if (profile.getTypes().contains(property.getType())) {
    divisible += 25; //5*5
    normProfile += 25;
    normProperty += 25;
}else{
    normProfile += 25;
}
```

У цьому лістингу був застосований коефіцієнт 5 через важливість атрибуту житла, divisible позначає скалярний добуток векторів, де у разі співпадіння 5*5 дорівнює 25, а у разі не співпадіння 5*0 дорівнює 0. NormProfile позначає норму профайлу та завжди дорівнює 5*5. NormProperty позначає норму житла та у разі співпадіння 5*5 дорівнює 25, а у разі не співпадіння 0*0 дорівнює 0. У даному випадку, при відсутності співпадіння немає сенсу додавати 0 до divisible та normProperty, тому 25 додається тільки для норми профілю.

Під час порівняння значень атрибуту «Опис житла» вони перетворюються у вектори за допомогою реалізації моделі sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 та порівнюються за класичною косинусною подібністю. Вектор опису житла порівнюється з усіма описами житла у профілі, і середнє значення цих порівнянь використовується як значення атрибуту житла у формулі. Значення профілю при цьому дорівнює 1. Використання цього підходу для атрибуту «Опис», можна побачити у лістингу 2.2.

Лістинг 2.2 – Розрахунок ступеня подібності значень атрибуту «Опис житла» (фрагмент файлу CosineSimilarity.java)

```
for (float[] tempVector : tempVectorsProfile) {
    double dotProduct = 0.0;
    double norm1 = 0.0;
    double norm2 = 0.0;
    for (int i = 0; i < tempVector.length; i++) {
        dotProduct += tempVector[i] * propertyVector[i];
        norm1 += tempVector[i] * tempVector[i];
        norm2 += propertyVector[i] * propertyVector[i];
    }
    norm1 = Math.sqrt(norm1);
    norm2 = Math.sqrt(norm2);
    tempCosDesc += dotProduct / (norm1 * norm2);
}
tempCosDesc = tempCosDesc/tempVectorsProfile.size();
divisible +=tempCosDesc;
normProperty += tempCosDesc*tempCosDesc;
normProfile += 1;
```

Для порівняння оснащення та додаткових параметрів використовується ваговий підхід: спершу обчислюється сумарна вага всіх елементів обладнання у профілі користувача, після чого визначається сумарна вага тих елементів, які присутні і в профілі, і в житлі. На основі цих значень обчислюється коефіцієнт подібності як частка збіжних ваг, що відображає, наскільки повно житло відповідає списку обладнання та додаткових параметрів профілю. Формулу цього розрахунку можна побачити нижче:

$$B_e = \frac{\sum(P \cap Q)}{\sum P}, \tag{2.2}$$

де B_e – коефіцієнт відповідності атрибуту equipment житла до equipment профілю;

$\sum P$ – сума всіх значень профілю;

$\sum(P \cap Q)$ – сума значень профілю, ключи яких є і у профілю, і у житла.

Отриманий коефіцієнт масштабується, після чого використовується у загальній формулі (2.1) як значення житла.

Використання цього підходу для обладнання та додаткових параметрів,

можна побачити у лістингу 2.3.

Лістинг 2.3 – Використання підходу розрахунку подібності для обладнання та додаткових параметрів (фрагмент файлу CosineSimilarity.java)

```
Map<Integer, Long> tempEquipmentsProfile =
profile.getEquipments();
Map<Integer, Long> tempEquipmentsProperty =
property.getEquipments();
long tempColEquipmentPoints = 0;
for (Long value : tempEquipmentsProfile.values()) {
    tempColEquipmentPoints += value;
}
long tempColPropertyPoints = 0;
for (Integer key : tempEquipmentsProperty.keySet()) {
    if (tempEquipmentsProfile.containsKey(key)) {
        tempColPropertyPoints +=
tempEquipmentsProfile.get(key);
    }
}
float tempEquipments = 1.0f - ((tempColEquipmentPoints -
tempColPropertyPoints) / (float) tempColEquipmentPoints);
tempEquipments *= 2;
divisible += tempEquipments*2;
normProfile += 4;
normProperty += tempEquipments*tempEquipments;
```

У цьому листингу визначається, наскільки список обладнання та додаткових параметрів об’єкта, який аналізується, відповідає списку профілю у межах від 0 до 1. Результат подвоюється та записується як значення для об’єкта, в той час як значення профілю дорівнює 2.

Рейтинг непереглянутих об’єктів житла, сформований з використанням метода косинусної подібності без урахування відповідності фільтрам, можна побачити у таблиці 2.3.

Таблиця 2.3 – Рейтинг непереглянутих об’єктів житла за косинусною подібністю без урахування відповідності фільтрам користувача

Порядковий номер у рейтингу	Ступень подібності об’єкта	Категорія	id об’єкта житла
1	0.9803737	Перша	3
2	0.9516129	Перша-друга	4

Продовження таблиці 2.3

Порядковий номер у рейтингу	Ступень подібності об’єкта	Категорія	id об’єкта житла
3	0.90336627	Перша-третя	5
4	0.73375815	Друга	8
5	0.7287298	Друга	6
6	0.71834534	Друга	7
7	0.7024365	Друга-третя	9
8	0.66235137	Третя	11
9	0.5505686	Третя	10
10	0.37787086	Третя	12

Як можна побачити, при використані косинусної подібності, усі непереглянуті об’єкти категорії «Перша» (з id 3, 4, 5) попали до перших п’яти місць побудованого рейтингу об’єктів житла, зайнявши 1,2 та 3 місця. Середня відповідність об’єктів тієї ж категорії дорівнює 0.94511762333, у той час як середня відповідність усіх непереглянутих об’єктів лише 0.730941349. Об’єкти тієї ж категорії являють собою 39% всієї важливості об’єктів.

Рейтинг непереглянутих об’єктів житла, сформований з урахуванням зменшення оцінки за фільтрами користувача наведено у таблиці 2.4.

Таблиця 2.4 – Рейтинг непереглянутих об’єктів житла за косинусною подібністю з урахуванням відповідності фільтрам користувача

Порядковий номер у рейтингу	Ступень подібності об’єкта	Категорія	id об’єкта житла
1	0.93741655	Перша	3
2	0.81680024	Перша-друга	4
3	0.7405913	Перша-третя	5
4	0.6183477	Друга	6
5	0.6015443	Друга	8
6	0.52283	Друга	7
7	0.42908517	Третя	11
8	0.3847403	Друга-третя	9
9	0.3693326	Третя	10

Продовження таблиці 2.4

Порядковий номер у рейтингу	Ступень подібності об’єкта	Категорія	id об’єкта житла
10	0.2389865	Третя	12

Як можна побачити, при використані косинусної подібності з урахуванням зменшення оцінки за фільтрами користувача, усі об’єкти тієї ж категорії попали до перших п’яти місць побудованого рейтингу об’єктів житла, зайнявши 1,2 та 3 місця. Середня відповідність об’єктів тієї ж категорії дорівнює 0.83160269666, у той час як середня відповідність усіх об’єктів лише 0.565967466. Об’єкти тієї ж категорії являють собою 44% всієї важливості об’єктів.

Вплив урахування фільтрів користувача на якість ранжування наведені у таблиці 2.5.

Таблиця 2.5 – Вплив урахування фільтрів користувача на якість ранжування

Показник	Базовий метод	З урахуванням фільтрів	Різниця
Кількість у топ-5 об'єктів, які є однокатегоріальними із переглянутими об'єктами	3 з 3	3 з 3	0
Позиції	1, 2, 3	1, 2, 3	0
Середній ступень подібності об'єктів, які є однокатегоріальними із переглянутими об'єктами	0.945	0.832	-11.96%
Середній ступень подібності непереглянутих об'єктів	0.731	0.566	-22.57%
Дискримінативна здатність	29%	47%	+18%

Дискримінативна здатність – здатність розрізняти об'єкти, явища або стимули за певними ознаками.

На основі результатів з таблиці 2.5 можна затверджувати, що

використання фільтрів збільшує ефективність методу косинусної подібності при розрахунку ступню подібності об’єктів житла.

2.1.3 Дослідження та перевірка методу евклідової відстані

Евклідова відстань – метрика в евклідовому просторі. Вона представляється як відстань між двома точками евклідового простору, що обчислюється за теоремою Піфагора. Наприклад, для векторів p та q евклідова відстань визначається наступним чином:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots (p_n - q_n)^2}\,, \qquad (2.3)$$

де d – відстань;

p, q – вектори;

p_n, q_n – координати точок [20].

Евклідова метрика – найбільш природна функція відстані, що виникає в геометрії, яка відображає інтуїтивні властивості відстані між точками.

Евклідова відстань є простим способом вимірювання відстані між об'єктами. Вона використовується в різних галузях для вирішення завдань, пов'язаних з простором і відстанню.

На відміну від косинусної подібності, яка враховує кут між двома точками або векторами, евклідова відстань фокусуватися на довжині лінії між ними. Також вона не обмежена відстанню від 0 до 1, замість цього більше значення позначає більшу відстань векторів.

Далі описуються деталі реалізації для тестування методу евклідової відстані.

Так, само, як і під час роботи з косинусною подібністю є числові значення (кількість кімнат, спальних місць, орендна плата, площа, поверх),

задані у вигляді діапазонів, тому під час порівняння кожної координати вектору береться те значення з діапазону профілю, яке найближче до відповідного значення вектору житла.

Приклад такого розрахунку з використанням вагового коефіцієнту, що дорівнює 3, наведений нижче у лістингу 2.4.

Лістинг 2.4 – Приклад розрахунку відстані при порівнянні з діапазоном та використанням коефіцієнту (фрагмент файлу EuclideanDistance.java)

```
IntSummaryStatistics tempProfileRooms = profile.getRooms();
int tempPropertyRooms = property.getColRooms();
//distance +=0-0=0;
if (tempProfileRooms.getMin() > tempPropertyRooms) {
    distance += (float) Math.pow((tempPropertyRooms-
tempProfileRooms.getMin())*3, 2);
}else if(tempProfileRooms.getMax() < tempPropertyRooms){
    distance += (float) Math.pow((tempPropertyRooms-
tempProfileRooms.getMax())*3, 2);
}
```

При роботі з категоріальними даними (місто, район, тип житла) використовується наступний підхід, якщо категорія житла є у листі категорій профілю відстань не змінюється, якщо такої категорії немає, відстань збільшується на відстань відповідну важливості параметру, приклад використання цього підходу можна побачити у лістингу 2.5.

Лістинг 2.5 – Приклад розрахунку відстані для категоріальних значень (фрагмент файлу EuclideanDistance.java)

```
if (!profile.getCities().contains(property.getCity())) {
    //0-0=0;
    distance = (float) Math.pow((0-30), 2);
};
```

Під час порівняння описів об’єктів житла вони перетворюються у вектори за допомогою реалізації моделі sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2, після чого порівнюються за класичним варіантом евклідової відстані. Вектор опису житла порівнюється з усіма описами житла, у профілі, і середнє значення цих порівнянь використовується як значення

житла у формулі. Значення профілю при цьому дорівнює 0. Використання цього підходу для значення атрибуту «Опис житла», можна побачити у лістингу 2.6.

Лістинг 2.6 – Розрахунок відстані для атрибуту «Опис» (фрагмент файлу EuclideanDistance.java)

```
List<float[]> tempVectorsProfile =
profile.getDescriptionVectors();
float[] propertyVector = property.getDescriptionVector();
if(!tempVectorsProfile.isEmpty() && propertyVector != null
&& propertyVector.length > 0){
    double tempEclidDesc = 0;
    for (float[] tempVector : tempVectorsProfile) {
        float radicand = 0;
        for (int i = 0; i < tempVector.length; i++) {
            radicand += (float) Math.pow((tempVector[i] -
propertyVector[i]), 2);
        }
        tempEclidDesc += Math.sqrt(radicand);
    }
    tempEclidDesc = tempEclidDesc/tempVectorsProfile.size();
    distance += (float) Math.pow((0-tempEclidDesc)*3, 2);
};
```

Для коректної роботи числових значень орендної плати був обраний відносний підхід, при якому різниця рахується не у стандартних числах, а у середній вартості житла профілю, після підрахунку отриманий результат помножується на коефіцієнт 7. Реалізацію даного розрахунку можна побачити у лістингу 2.7.

Лістинг 2.7 – Розрахунок відстані для атрибуту «Орендна плата» (фрагмент файлу EuclideanDistance.java)

```
IntSummaryStatistics tempProfilePrice = profile.getPrice();
int tempPropertyPrice = property.getPrice();
//0-0=0;
if (tempProfilePrice.getMin() > tempPropertyPrice) {
    float tempPriceMin = (((tempProfilePrice.getMin() -
tempPropertyPrice)) / (float)
tempProfilePrice.getAverage())*7;
    distance += ((float) Math.pow(0-tempPriceMin, 2));
}else if(tempProfilePrice.getMax() < tempPropertyPrice){
```

```
    float tempPriceMax = ((tempPropertyPrice -
tempProfilePrice.getMax()) / (float)
tempProfilePrice.getAverage())*7;
    distance += ((float) Math.pow(0-tempPriceMax, 2));
}
```

Для порівняння параметрів обладнання та додаткових параметрів спочатку підраховується загальна кількість балів обладнання в профілі. Потім підсумовуються бали тільки тих елементів, які присутні і в профілі, і у об’єкті житла. Ці числа використовуються у фінальній формулі. Використання цього підходу для обладнання та додаткових параметрів, можна побачити у лістингу 2.8.

Лістинг 2.8 – Розрахунок відстані для обладнання та додаткових параметрів (фрагмент файлу EuclideanDistance.java)

```
Map<Integer, Long> tempEquipmentsProfile =
profile.getEquipments();
Map<Integer, Long> tempEquipmentsProperty =
property.getEquipments();
long tempColEquipmentPoints = 0;
for (Long value : tempEquipmentsProfile.values()) {
    tempColEquipmentPoints += value;
}
long tempColPropertyPoints = 0;
for (Integer key : tempEquipmentsProperty.keySet()) {
    if (tempEquipmentsProfile.containsKey(key)) {
        tempColPropertyPoints +=
tempEquipmentsProfile.get(key);
    }
}
distance += (float) Math.pow(tempColEquipmentPoints-
tempColPropertyPoints, 2);
```

Рейтинг непереглянутих об’єктів житла, сформований з використанням евклідової відстані, можна побачити у таблиці 2.6.

Таблиця 2.6 – Рейтинг непереглянутих об’єктів житла за евклідовою відстанню

Порядковий номер у рейтингу	Евклідова відстань	Категорія об’єкта	Id об’єкта
1	11.216299	Перша	3
2	21.988783	Перша-третя	5
3	22.321035	Перша-друга	4
4	31.994844	Друга	6
5	32.317074	Друга	8
6	33.53902	Друга	7
7	33.568184	Третя	11
8	34.97311	Друга-третя	9
9	45.392124	Третя	12
10	46.411488	Третя	10

Як можна побачити, при використані евклідової відстані, усі об’єкти тієї ж категорії попали до перших п’яти місць створеного рейтингу житла, зайнявши 1, 2 та 3 місця. Середня відстань об’єктів тієї ж категорії дорівнює 18.5087056667, у той час як середня відстань усіх об’єктів 31.3721961. Об’єкти тієї ж категорії мають середню відстань на 41% менше ніж усі непереглянуті об’єкти.

Рейтинг непереглянутих об’єктів житла, сформований з використанням евклідової відстані та урахуванням зменшення оцінки за фільтрами (фінальний) користувача наведено у таблиці 2.7.

Таблиця 2.7 – Рейтинг непереглянутих об’єктів житла за евклідовою відстанню з урахуванням фільтрів користувача

Порядковий номер у рейтингу	Евклідова відстань	Категорія об’єкта	Id об’єкта
1	12.716299	Перша	3
2	26.571035	Перша-друга	4
3	27.38554	Перша-третя	5
4	40.513206	Друга	6

Продовження таблиці 2.7.

Порядковий номер у рейтингу	Евклідова відстань	Категорія об’єкта	Id об’єкта
5	41.556934	Друга	8
6	42.28902	Друга	7
7	43.692413	Третя	11
8	43.886024	Друга-третя	9
9	56.546986	Третя	10
10	57.937958	Третя	12

Як можна побачити, при використані евклідової відстані, усі об’єкти тієї ж категорії попали до перших п’яти місць створеного рейтингу житла, зайнявши 1, 2 та 3 місця. Середня відстань об’єктів тієї ж категорії дорівнює 22.2242913333, у той час як середня відстань усіх об’єктів 39.3095415. Об’єкти тієї ж категорії мають середню відстань на 43% менше ніж усі непереглянуті об’єкти.

Вплив урахування фільтрів користувача на якість ранжування наведені у таблиці 2.8.

Таблиця 2.8 – Вплив урахування фільтрів користувача на якість ранжування

Показник	Базовий метод	З фільтрами	Різниця
Кількість у топ-5 об'єктів, які є однокатегоріальними із переглянутими об'єктами	3 з 3	3 з 3	0
Позиції об'єктів, які є однокатегоріальними із переглянутими об'єктами, у рейтингу	1, 2, 3	1, 2, 3	0
Середня відстань для об'єктів, які є однокатегоріальними із переглянутими об'єктами	18.509	22.224	+20%
Середня відстань для непереглянутих об’єктів	31.372	39.31	+25%
Дискримінативна здатність	41%	43%	+2%

На основі результатів з таблиці 2.8 можна затверджувати, що врахування використаних фільтрів при розрахунку евклідової відстані підвищує дискримінативну здатність методу.

2.1.4 Дослідження та перевірка методу відстані Гауера

Відстань Гауера – це метод, який дозволяє вимірювати «відстань» між об'єктами, у яких ознаки можуть бути числові та категоріальні. Ключовою особливість методу є нормалізація кожної ознаки відповідно максимального та мінімального значення точки серед векторів для числових значень. Дистанція обчислюється наступним чином для кожної ознаки знаходять її нормовану відстань, потім усі отримані відстані підсумовують і діляться на кількість ознак, які порівнювали.

Традиційні метрики, такі як евклідова відстань, не можуть у чистому виді обробляти категоріальні дані. Наприклад, у категоріальних атрибутах, таких як «Тип житла» або «Місто», між категоріями немає внутрішньої «відстані». Відстань Гауера враховує це, надаючи простий, але потужний спосіб обчислення відстаней, що включає обидва типи даних в єдиній структурі.

Відстань Гауера поєднує різні метрики відстані для кожного типу атрибуту і обчислює загальну відстань між двома точками як середнє значення індивідуальних відстаней атрибутів.

Для числових атрибутів ми розраховуємо нормалізовану різницю:

$$d = \frac{|x_i - x_j|}{R}, \tag{2.4}$$

де d – відстань;

x_i, x_j – числові значення;

R – різниця між максимальним на мінімальним значенням атрибуту.

Для категоріальних значень:

$$d = \begin{cases} 0 \; if \; x_i = x_j \\ 1 \; if \; x_i \neq x_j \end{cases}, \tag{2.5}$$

де d – відстань;

x_i, x_j– категоріальні значення.

Для остаточного розрахунку відстані Гауера використовується наступна формула:

$$D(i,j) = \frac{1}{p}\sum_{k=1}^{p} d_{i,j}^{k} , \tag{2.6}$$

де D – відстань між двома точками даних;

p – кількість елементів;

k – номер елементу;

d – відстань між двома одномірними елементами різних точок даних [21].

Відстань Гауера – це потужний метод для вимірювання відмінностей між окремими елементами в наборах даних із змішаними типами даних. Він забезпечує надійне рішення там, де традиційні показники виявляються недостатніми, що робить його ідеальним для використання у кластеризації, системах рекомендацій та сегментації клієнтів. Обробляючи як числові, так і категоріальні змінні, він гарантує, що всі аспекти даних будуть враховані.

Реалізація відстані Гауера для порівняння об’єктів житла, на відміну від косинусної подібності та евклідової відстані, фактично не потребувала якихось змін у формулі, єдине, що були додані коефіцієнти які піднімають вплив міста та понижують вплив поверху на фінальний результат.

Рейтинг об’єктів житла, сформований з використанням відстані Гауера та без урахування фільтрів користувача можна побачити у таблиці 2.9.

Таблиця 2.9 – Рейтинг житла за відстанню Гауера без урахування фільтрів користувача

Порядковий номер у рейтингу	Відстань Гауера	Категорія об’єкта	Id об’єкта
1	0.15962613	Перша	3
2	0.2850607	Перша-друга	4
3	0.3295318	Перша-третя	5
4	0.3922969	Друга	8
5	0.40111235	Друга	6
6	0.41375095	Друга	7
7	0.46404877	Друга-третя	9
8	0.50908226	Третя	11
9	0.65810627	Третя	10
10	0.72134596	Третя	12

Як можна побачити, при використані відстані Гауера, усі об’єкти тієї ж категорії попали до перших п’яти місць створеного рейтингу житла, зайнявши 1, 2 та 3 місця. Середня відстань об’єктів тієї ж категорії дорівнює 0.25807287666, у той час як середня відстань усіх об’єктів 0.433396209. Об’єкти тієї ж категорії мають середню відстань на 40% менше ніж усі непереглянуті об’єкти.

Рейтинг житла сформований з використанням відстані Гауера та урахуванням зменшення оцінки за фільтрами користувача наведений у таблиці 2.10.

Як можна побачити, при використані відстані Гауера з використанням фільтрів усі об’єкти тієї ж категорії попали до перших п’яти місць створеного рейтингу житла, зайнявши 1, 2 та 3 місця. Середня відстань об’єктів тієї ж категорії дорівнює 0.29557287333, у той час як середня відстань усіх об’єктів 0.522562869. Об’єкти тієї ж категорії мають середню відстань на 43% менше ніж усі непереглянуті об’єкти.

Таблиця 2.10 – Рейтинг житла за відстанню Гауера та урахуванням зменшення оцінки за фільтрами користувача

Порядковий номер у рейтингу	Відстань Гауера	Категорія об’єкта	Id об’єкта
1	0.17212613	Перша	3
2	0.32672736	Перша-друга	4
3	0.38786513	Перша-третя	5
4	0.46361235	Друга	6
5	0.47563022	Друга	8
6	0.48458427	Друга	7
7	0.56404877	Друга-третя	9
8	0.6486656	Третя	11
9	0.79560626	Третя	10
10	0.9067626	Третя	12

Вплив урахування фільтрів користувача на якість розрахунку ступеня подібності наведені у таблиці 2.11.

Таблиця 2.11 – Вплив урахування фільтрів користувача на якість ранжування

Показник	Базовий метод	З фільтрами	Різниця
Кількість у топ-5 об'єктів, які є однокатегоріальними із переглянутими об'єктами	3 з 3	3 з 3	0
Позиції об'єктів, які є однокатегоріальними із переглянутими об'єктами, у рейтингу"	1, 2, 3	1, 2, 3	0
Середня відстань для об'єктів, які є однокатегоріальними із переглянутими об'єктами	0.258	0.296	+15%
Середня відстань для непереглянутих об’єктів	0.433	0.523	+21%
Дискримінативна здатність	40%	43%	+3%

На основі результатів з таблиці 2.11 можна затверджувати, що

використання фільтрів при використанні евклідової відстані також, як і при використанні вже розглянутих методів, збільшує ефективність відбору релевантних профілю об’єктів житла.

2.1.5 Порівняння методів розрахунку ступеня подібності об'єктів

Проведене дослідження продемонструвало, що всі три методи обчислення ступеня подібності об'єктів показують схожі результати за умови належного налаштування. Ключовим фактором ефективності виявилося правильне визначення ваг і коефіцієнтів для кожного методу, що забезпечило розміщення однокатегоріальних об'єктів на перших позиціях у всіх випадках. Водночас косинусна подібність продемонструвала дещо кращі показники порівняно з розглянутими альтернативними підходами. Порівняльні характеристики методів розрахунку ступеня подібності можна побачити у таблиці 2.12.

Таблиця 2.12 – Порівняльні характеристики методів розрахунку ступеня подібності об'єктів

Метод / Критерій	Косинусна подібність	Евклідова відстань	Відстань Гауера
Позиції об'єктів, які є однокатегоріальними із переглянутими об'єктами, у рейтингу	1-3 позиції	1-3 позиції	1-3 позиції
Дискримінативна здатність	47%	43%	43%
Підтримка категоріальних значень	З доробкою	З доробкою	Так
Вразливість до екстра значень	Висока	Висока	Низька
Потреба в налаштуванні коефіцієнтів важливості атрибутів	Висока	Висока	Низька
Кількість переваг	4,5	3,5	4

Кожен досліджений метод має певні переваги та обмеження.

Відстань Гауера без додаткових модифікацій здатна працювати з категоріальними змінними та містить вбудовану нормалізацію параметрів, яка виявилася настільки ефективною, що практично не потребувала використання ваг під час тестування. Проте залежність цього методу нормалізації від найбільшого та найменшого значень серед векторів, що порівнюються, робить його вразливим до зменшення значення выдстані атрибутів при наявності навіть хоча б одного значення з аномальною величиною.

Косинусна подібність та евклідова відстань, навпаки, потребують розробки власних механізмів нормалізації та обробки категоріальних змінних, але демонструють більшу стійкість до аномальних значень. Варто відзначити, що налаштування нормалізації та ваг для евклідової відстані є більш інтуїтивним завдяки прозорішій інтерпретації цього методу.

Для реалізації модуля було обрано модифіковану косинусну подібність як оптимальний метод побудови рекомендацій щодо вибору нерухомості. Це рішення обґрунтовується вищою стабільністю порівняно з відстанню Гауера та кращими результатами експериментального тестування. Для обробки категоріальних змінних до методу інтегровано підхід, аналогічний методу відстані Гауера, а також прийнято використання системи ваг відповідно до значущості атрибутів.

2.2 Вибір методу перетворення значень текстових атрибутів об’єктів на вектори

2.2.1 Підхід до вибору кращого методу для перетворення текстових атрибутів об’єктів на вектори

Оцінка методів для перетворення текстових атрибутів об’єктів на вектори у контексті рекомендаційних систем буде проводитися за зовнішньою

оцінкою (Extrinsic Evaluation), тобто вимірюванням ефективності методів при використанні в реальних завданнях, що дає більш практичне уявлення про їх корисність. Для експериментального порівняння методів ембедингу були обрані методи Bag-of-Words, TF-IDF та реалізація нейронної моделі sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2, яка є моделлю класу Sentence-BERT (архітектурна модіфікація BERT).

Для проведення експерименту були використані ті ж самі об’єкти житла, що і при дослідженні методів розрахунку ступеня подібності. Але замість об’єктів житла порівнювалися значення текстового атрибуту "Опис об'єкту житла". Довжина описів складала від 30 до 40 слів. Два тексти категорії «Перша» були відмічені як переглянуті, і усі інші тексти перевірялись на подібність до них.

При виконанні експерименту методи ембедингу були використані для перетворення текстів на вектори, а обраний раніше метод розрахунку ступеня подібності (метод косинусної подібності) – для визначення подібності об'єктів.

При виборі кращого методу враховувалися критерії, що представлені у таблиці 2.13.

Таблиця 2.13 – Критерії оцінювання методів перетворення текстових атрибутів на вектори

Критерій	Опис критерія
Позиції об'єктів, які є однокатегоріальними із переглянутими об'єктами, у рейтингу	Це легкий та інтуїтивний спосіб виявити відповідність результатів
Дискримінативна здатність (в %)	Для виявлення наскільки метод відрізняє однокатегоріальні об’єкти від інших

2.2.2 Дослідження та перевірка BERT-подібної моделі

Реалізація нейронної моделі sentence-transformers/paraphrase-

multilingual-MiniLM-L12-v2 є моделлю класу Sentence-BERT. Sentence-BERT є архітектурною модіфікацією BERT.

BERT (Bidirectional Encoder Representations from Transformers) – це провідна модель для роботи з природною мовою, яка базується на архітектурі трансформера. Ця нейромережева структура застосовує механізм уваги для опрацювання та інтерпретації текстових даних.

Механізм уваги при обробці природної мови (NLP) дає моделі змогу під час аналізу акцентувати увагу на найвагоміших сегментах тексту, для його кращого розуміння.

Для перевірки метода BERT була обрана модель «paraphrase-multilingual-MiniLM-L12-v2». Це багатомовна модель векторизації речень, вона відображає тексти в 384-вимірному щільному векторному просторі і може використовуватися для таких завдань, як порівняння або семантичний пошук. Для використання цієї моделі був вибраний Java-фреймворк Deep Java Library (DJL), призначений для роботи з моделями машинного навчання, який забезпечує можливість завантаженнята застосування моделей безпосередньо в Java-середовищі без потреби ручної взаємодії з Python-екосистемою.

При роботі спочатку потрібно загрузити модель, як представлено у лістингу 2.9.

Лістинг 2.9 – Код завантаження моделі paraphrase-multilingual-MiniLM-L12-v2 (фрагмент з файлу BertTextToVector.java)

```
Criteria<String, float[]> criteria = Criteria.builder()
        .setTypes(String.class, float[].class)

.optModelUrls("djl://ai.djl.huggingface.pytorch/sentence -
transformers/paraphrase-multilingual-MiniLM-L12-v2")
        .optEngine("PyTorch")
        .optProgress(new ProgressBar())
        .build();

this.model = criteria.loadModel();
this.predictor = model.newPredictor();
```

Заголом векторизиція тут працює так, при створені об'єкту

BertTextToVector, у конструкторі відбувається завантаження моделі. DJL завантажує з Hugging Face файли нейромережі BERT. PyTorch ініціалізує всю структуру нейромережі. З завантаженої моделі створюється predictor, який буде керувати процесом ембедінгу.

При виконанні predictor.predict(), текст підготавлюється для нейромережі, тому що BERT не розуміє слова безпосередньо. Токенайзер розбиває фразу на маленькі частини (subword tokens) і перетворює кожну частину в числовий ID згідно з вбудованим словником. Ці числа упаковуються в тензор, спеціальну структуру даних. Також створюється attention mask (маска уваги), яка показує, які токени справжні, а які додані для вирівнювання довжини. Все це відправляється в BERT.

Всередині BERT, перший шар (embedding) перетворює кожен числовий ID у вектор із 384 чисел, початкове представлення токена. Далі ці вектори проходять через 12 однакових блоків трансформера. У кожному блоці є механізм self-attention, де обчислюється для кожного токена вага важливості всіх інших токенів, потім формується новий вектор як зважена сума їх представлень. Після attention йдуть feed-forward шари, які роблять нелінійні перетворення, збагачуючи уявлення. Кожен з 12 блоків робить вектори все більш «розумними», насичуючи їх контекстом і змістом.

На виході з останнього шару BERT отримується матриця векторів, по одному вектору на кожен токен. Далі застосовується pooling, вибір одного вектора з матриці вихідних векторів BERT, зазвичай береться вектор спеціального токена (CLS) з першої позиції.

Отриманий вектор нормалізується (зазвичай діленням на його довжину), щоб усі вектори були порівнянні за масштабом. Нарешті, тензор конвертується з внутрішнього формату PyTorch у звичайний Java масив float[] та Predictor повертає масив із 384 чисел.

Рейтинг текстів сформований з використанням BERT та косинусної подібності можна побачити нижче, у таблиці 2.14.

Таблиця 2.14 – Рейтинг текстів сформований з використанням BERT-подібної моделі та косинусної подібності

Порядковий номер у рейтингу	Ступень подібності	Категорія об’єкта
1	0.7691437304019928	Перша
2	0.7372039556503296	Перша-третя
3	0.7170676589012146	Перша-друга
4	0.6820610463619232	Друга
5	0.6545446813106537	Друга-третя
6	0.65058434009552	Друга
7	0.6341002583503723	Третя
8	0.5926044881343842	Друга
9	0.5743103325366974	Третя
10	0.5270087420940399	Третя

При використані BERT-подібної моделі з косинусною подібністю, усі текстові описи непереглянутих об’єктів категорії «Перша» попали до перших трьох місць створеного рейтингу. Середня подібність текстів тієї ж категорії дорівнює 0.74113844831, у той час, як середня подібність усіх непереглянутих текстів – лише 0.65386292338. Тексти тієї ж категорії мають середню подібність на 13% більше ніж непереглянуті тексти.

2.2.3 Дослідження та перевірка методу Bag-of-words

Bag of Words був одним із ранніх практичних методів, який дозволяв подати текст у вигляді чисел. Цей метод реалізує принцип BoW, який полягає в тому, що текст перетворюється на вектор тексту, де кожний елемент показує, скільки разів певне слово зустрічається в документі.

Механізм методу такий: спочатку формується корпус або словник усіх унікальних слів із наявних текстів. Потім кожен текст подається у вигляді

вектору, розмірність якого дорівнює розміру словника, а значення кожної розмірності дорівнює частоті вживання слова в тексті.

Реалізація створення словника усіх унікальних слів наведена у лістингу 2.10.

Лістинг 2.10 – Реалізація створення словника усіх унікальних слів (фрагмент з файлу BagOfWords.java)

```
Set<String> globalVocabulary = new LinkedHashSet<>();
for(Text text : allTexts){
    String[] tokens =
text.getText().toLowerCase().split("\\P{L}+");
    globalVocabulary.addAll(Arrays.asList(tokens));
}
```

Реалізація створення вектору тексту представлена у лістингу 2.11.

Лістинг 2.11 – Реалізація створення вектору тексту (фрагмент з файлу BagOfWords.java)

```
for(Text text:allTexts){
    String[] tokens =
text.getText().toLowerCase().split("\\P{L}+");
    Map<String, Double> vector = new LinkedHashMap<>();
    for (String word : globalVocabulary) {
        vector.put(word, 0.0);
    }
    for (String token : tokens) {
        vector.put(token, vector.get(token) + 1);
    }
    text.setVectorBOWIT(vector);
}
```

У даному коді для кожного тексту проходить токенізація, створення вектора признаків та підлік частоти слів, після чого отриманий результат зберегається у VectorBOWIT.

Рейтинг текстів сформований з використанням Bag of Words та косинусної подібності можна побачити нижче, у таблиці 2.15.

Таблиця 2.15 – Рейтинг текстів сформований з використанням Bag of Words та косинусної подібності.

Порядковий номер у рейтингу	Ступень подібності	Категорія об’єкта
1	0.26165946831484665	Перша
2	0.24094262117709064	Друга-третя
3	0.23326851658421704	Перша-третя
4	0.18625362947141832	Третя
5	0.14259438353340448	Друга
6	0.13640689825271052	Перша-друга
7	0.11750982553894372	Третя
8	0.06658014343186634	Друга
9	0.06089070039196951	Друга
10	0.05822264572071275	Третя

При використані Bag of Words з косинусною подібністю тільки два тексти непереглянутих об’єктів категорії «Перша» з трьох попали до перших п’яти місць рейтингу, зайнявши перше та третє місце. Середня подібність текстів категорії «Перша» дорівнює 0.21044496105 у той час, як середня подібність усіх непереглянутих текстів лише 0.15043288324. Тексти категорії «Перша» мають середню подібність на 40% більше ніж непереглянуті тексти, хоча у даному випадку це свідчіть про великий розрив між першими та останніми місцями.

2.2.4 Дослідження та перевірка методу TF-IDF

Метод TF-IDF працює схожим з Bag of Words чином, але, додатково, знижує важливість часто вживаних серед усіх текстів слів, це викликано знизити важливість таких слів як «і», «та», «або» тощо. Таким чином TF-IDF приділяє більше уваги словам, що виділяють речення та несуть його сенс.

Міра важливості слова в TF-IDF визначається наступним чином частота терміну в документі (TF) множиться на зворотна частоту документа (IDF).

Код, що підраховує IDF представлений у лістингу 2.12.

Лістинг 2.12 – Підрахунок IDF (фрагмент з файлу TfIdf.java)

```
Map<String, Double> idf = new HashMap<>();
for(Text text:allTexts){
    Set<String> uniqueWords = new HashSet<>(

Arrays.asList(text.getText().toLowerCase().split("\\s+"))
    );
    for (String word : uniqueWords) {
        idf.put(word, idf.getOrDefault(word, 0.0) + 1);
    }
}
for (String word : idf.keySet()) {
    double df = idf.get(word);
    idf.put(word, Math.log((1.0 + allTexts.size()) / (1.0 +
df)) + 1.0);
}
```

Розрахунок TF та TF-IDF представлений нижче, у лістингу 2.13.

Лістинг 2.13 – Розрахунок TF та TF-IDF (фрагмент з файлу TfIdf.java)

```
for (Text text : allTexts) {
    String doc = text.getText();
    String[] words = doc.toLowerCase().split("\\s+");
    Map<String, Integer> wordCount = new HashMap<>();

    for (String word : words) {
        wordCount.put(word, wordCount.getOrDefault(word, 0)
+ 1);
    }

    Map<String, Double> tfidf = new HashMap<>();
    for (Map.Entry<String, Integer> entry :
wordCount.entrySet()) {
        String term = entry.getKey();
        double tf = (double) entry.getValue() /
words.length;
        tfidf.put(term, tf * idf.get(term));
    }
    text.setVectorBOWIT(tfidf);

}
```

Рейтинг текстів сформований з використанням TF-IDF та косинусної подібності можна побачити нижче, у таблиці 2.16.

Таблиця 2.16 – Рейтинг текстів сформований з використанням TF-IDF та косинусної подібності.

Порядковий номер у рейтингу	Ступень подібності	Категорія об’єкта
1	0.10174167303283582	Перша
2	0.0785689789606471	Перша-третя
3	0.07061774022905887	Друга-третя
4	0.0645699560811784	Третя
5	0.04855037015690574	Перша-друга
6	0.03172649726679427	Друга
7	0.029437279870351878	Третя
8	0.02140441716424031	Друга
9	0.02008406200310641	Третя
10	0.0188169037405954	Друга

При використані TF-IDF з косинусною подібністю, усі три тексти непереглянутих об’єктів категорії «Перша» попали до перших п’яти місць створеного рейтингу, зайнявши перше, друге та п’яте місце. Середня подібність текстів категорії «Перша» дорівнює 0.07628700738, у той час як середня подібність усіх непереглянутих текстів лише 0.04855178785. Непереглянуті тексти категорії «Перша» мають середню подібність на 57% більше ніж усі непереглянуті тексти, що визвано великим розривом серед перших та останніх місць рейтингу.

2.2.5 Порівняння методів для перетворення текстових атрибутів об’єктів на вектори

Проведене дослідження продемонструвало, що лише BERT зміг повністю справитися з завданнями. TF-IDF зміг видати задовільний результат, у той час як Bag of Words видав найгірший результат серед усіх методів. Порівняльні характеристики методів, які перевірялися, наведені у таблиці 2.17.

Таблиця 2.17 – Порівняльні характеристики методів для перетворення текстових атрибутів об’єктів на вектори

Критерій \ Метод	BERT-подібна модель	TF-IDF	Bag of Words
Позиції об'єктів, які є однокатегоріальними із переглянутими об'єктами, у рейтингу	1-3 позиції	1,2 та 5 позиції	1,3 та 6 позиції
Дискримінативна здатність	13%	57%	40%
Кількість переваг	1	1	0

Дані таблиці показують, що безсумнівним кращім рішенням щодо вибору методу ембедингу буде метод BERT. Хоча інші методи мають більшу дискримінативну здатність, позиції текстів непереглянутих об’єктів категорії «Перша» в рейтингу свідчать, тільки, про великий розрив серед перших та останніх місць рейтингу.

Отже, на основі отриманих результатів можна зробити висновок, що використання BERT-подібної моделі є кращим способом перетворення текстових атрибутів об’єктів на вектори для побудови рекомендацій клієнтам в ІС агентства нерухомості.

3 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОБУДОВИ РЕКОМЕНДАЦІЙ КЛІЄНТАМ В ІНФОРМАЦІЙНІЙ СИСТЕМІ АГЕНТСТВА НЕРУХОМОСТІ

3.1 Опис інформаційної технології побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості

В рамках роботи була розроблена ІТ побудови рекомендацій клієнтам в ІС агентства нерухомості. Для опису ІТ розроблено модель типу IDEF0, яка представлена на рисунку 3.1.

На цій діаграмі відображені всі підпроцеси автоматизованого процесу побудови рекомендацій клієнтам в ІС агентства нерухомості та їх вхідні, вихідні дані, механізми та управління.

При формуванні рекомендацій збираються запити користувача за допомогою модуля пошуку та перегляду житла та на основі них з використанням модуля рекомендацій, бази даних та реалізації BERT-подібної моделі формується електронний лист з рекомендаціями для клієнта. Цей процес відбувається відповідно до закону України «Про захист персональних даних», політики конфіденційності ІС та content-based підходу.

На даній діаграмі, яка описує ІТ, представлено сім основних етапів.

Перший етап – це збирання інформації про дії користувача. Під час виконання цього етапу фіксуються об’єкти житла, які переглядав користувач та фільтри, які він задавав у пошуку. Вхідними даними етапу є запити користувача до системи та інформація про користувача. Керуючими елементами – політика конфіденційності та Закон України про захист персональних даних, які регламентують правила збору, обробки та зберігання користувацьких даних. До механізмів етапу відносяться модуль пошуку та перегляду житла та база даних ІС. Вихідними даними етапу є інформація про дії користувача (перегляди об’єктів житла та фільтри, що задавав користувач при пошуку житла).

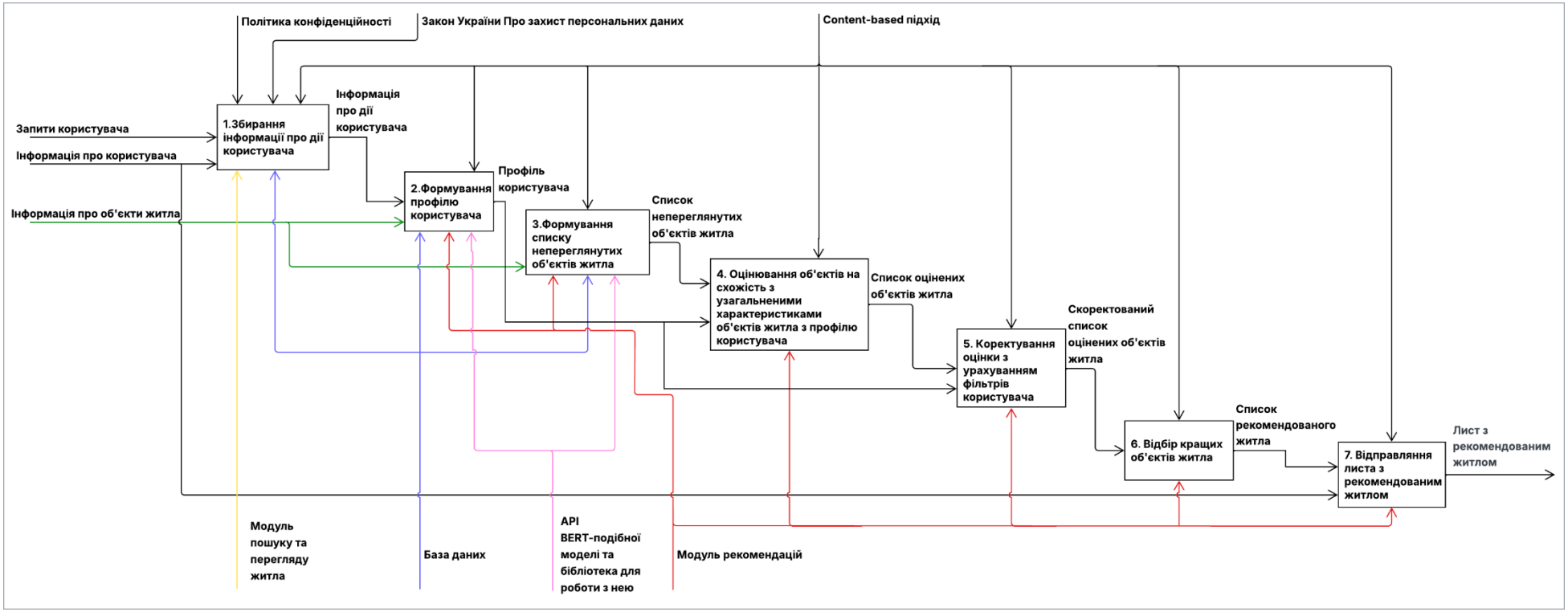


Рисунок 3.1 – Модель IDEF0 інформаційної технології побудови рекомендацій клієнтам в ІС агентства нерухомості

Другий етап – це формування профілю користувача. На цьому етапі на основі зібраної інформації формується профіль користувача, що містить інформацію про узагальнені атрибути житла, яке він переглядав та фільтри, які він задавав. Вхідними даними етапу є інформація про дії користувача та інформація про об'єкти житла. Керуючим елементом є content-based підхід до формування рекомендацій. До механізмів етапу відносяться база даних, модуль рекомендацій, модель та бібліотека для роботи з BERT. Вихідними даними етапу є сформований профіль користувача.

Третій етап – це формування списку непереглянутих об'єктів житла. Під час виконання цього етапу відбираються об’єкти на сторінки яких не заходив користувач та з них формується список. Вхідними даними етапу є профіль користувача. Керуючим елементам content-based підхід. До механізмів етапу відносяться база даних, модуль рекомендацій, модель та бібліотека для роботи з BERT. Вихідними даними етапу є список непереглянутих об'єктів житла.

Четвертий етап – оцінювання об'єктів на схожість з узагальненими характеристиками об'єктів житла з профілю користувача. Під час виконання цього етапу з використанням методів косинусної подібності та BERT-подібної моделі формується рейтинг житла за подібністю до узагальнених атрибутів житла з профілю користувача. Вхідними даними етапу є список непереглянутих об'єктів житла. Керуючим елементом content-based підхід. Механізмом етапу є модуль рекомендацій. Вихідними даними етапу є список оцінених об'єктів житла.

П’ятий етап – це коректування оцінки з урахуванням фільтрів користувача. На цьому етапі оцінки об’єктів житла знижуються у разі, якщо вони суперечать фільтрам, які задавав користувач, ступень зменшення залежить від ступеня суперечності фільтрам. Вхідними даними етапу є список оцінених об'єктів житла. Керуючим елементом content-based підхід. До механізмів етапу відносяться модуль рекомендацій. Вихідними даними етапу є скоректований список оцінених об'єктів житла.

Шостий етап – це відбір найбільш підходящих об'єктів житла. Під час

його виконання обираються п’ять об’єктів з кращими оцінками подібності та на їх основі формується електронний лист з рекомендованим житлом. Вхідними даними етапу є скоректований список оцінених об'єктів житла. Керуючим елементом content-based підхід. До механізмів етапу відносяться модуль рекомендацій. Вихідними даними етапу є список рекомендованого житла

Сьомий етап – це відправляння електронного листа з рекомендованим житлом. На цьому етапі формується електронний лист з інформацією та посиланням на п’ять найбільш відповідних об’єктів житла та надсилається на пошту користувачу. Для надсилання електронного листа клієнту має використовуватися поштовий сервіс. Вхідними даними етапу є список рекомендованого житла. Керуючим елементом content-based підхід. До механізмів етапу відносяться модуль рекомендацій та поштовий сервіс. Вихідними даними етапу є електронний лист з рекомендованим житлом.

3.2 Опис впровадження інформаційної технології побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості

Інформаційна технологія буде впроваджена через реалізацію модуля рекомендацій у ІС. Для розробки модуля було обране середовище розробки IntelliJ IDEA. До плюсів IntelliJ IDEA можна віднести:

а) IntelliJ IDEA підтримує безліч мов і фреймворків, але головне він підтимує мовуJava, на якій буде написаний модуль;

б) IntelliJ IDEA має відмінні інструменти рефакторингу, які дозволяють безпечно змінювати структуру коду без ризику зламати проект. Наприклад, при перейменуванні методу або класу IDE автоматично оновлює всі пов'язані місця;

в) IntelliJ IDEA має автодоповнювач коду, який з контексту, пропонує

методи, змінні та типи даних;

г) IntelliJ IDEA дозволяє швидко переходити до класів, методів, використання змінних, реалізації інтерфейсів.

Для роботи з базою даних був обраний MySQL Workbench через такі переваги:

а) MySQL Workbench це офіційний інструмент від Oracle, який має повну сумісність з MySQL і отримує регулярні оновлення;

б) MySQL Workbench має зручний та звичний візуальний інтерфейс, який дозволяє робототи з базами даних, таблицями та зв'язками без необхідності писати усе вручну;

в) MySQL Workbench є безкоштовним, і всі його функцій доступні без придбання ліцензії.

Модуль побудови рекомендацій написаний на Java. Головною причиною такого рішення є те, що програмне забезпечення ІС, для якої розробляється модуль, написаний на Java, але окрім цього Java є надійним та безпечним варіантом з великою спільнотою користувачів, високою продуктивністю, зручний для створення масштабованих та підтримуваних проектів та має гарне середовище розробки у виді IntelliJ IDEA.

Як BERT-модель має використовуватися реалізація нейронної моделі sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2. Це багатомовна модель для отримання векторних представлень речень. Вона навчена так, щоб речення зі схожим змістом мали близькі вектори, навіть якщо вони написані різними мовами, що є поїзною рисою у специфіці українського ринку оренди. Для використання цієї моделі має бути застосоване API, тому використовується Deep Java Library.

Deep Java Library – це, відкрита бібліотека для роботи зі штучним інтелектом і машинним навчанням на Java. Вона надає зручний API для завантаження, навчання та використання моделей машинного навчання, підтримує кілька популярних AI-двигунів і дозволяє вбудовувати інтелектуальні функції безпосередньо в Java-додатки.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ПОБУДОВИ РЕКОМЕНДАЦІЙ ТА ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА НАУКОВИХ РЕЗУЛЬТАТІВ

4.1 Реалізація модуля побудови рекомендацій в інформаційній системі агентства нерухомості

Розроблена ІТ має бути реалізована в ІС агенства нерухомості за допомогою модуля побудови рекомендацій.

Модуль має монолітну архітектуру з архітектурним шаблоном Model-View-Controller. Монолітна архітектура – це класичний метод побудови програмного забезпечення, коли всі складові системи: користувацький інтерфейс, бізнес-логіка та логіка обробки даних, об'єднані в один програмний код і розгортаються разом. Така структура працює як єдине ціле, де всі частини використовують спільні ресурси, пам'ять і обчислювальну потужність.

Така архітектура, зазвичай, реалізується через багаторівневу модель, у якій рівень користувацького інтерфейсу, рівень бізнес-логіки та рівень доступу до даних структуровані горизонтально із визначеною ієрархією залежностей. Робота всіх модулів у спільному контексті виконання забезпечує зручність тестування функціоналу.

Монолітні системи зазвичай будуються за відомими корпоративними шаблонами, такими як Model-View-Controller MVC або багаторівневі архітектури, які розділяють функції, але залишаються простими у впровадженні. Зміна будь-якої частини моноліту означає, що треба оновлювати весь додаток цілком — це забезпечує узгодженість, але ускладнює часті оновлення. Монолітний підхід полегшує початок розробки, бо компоненти не потребують складної взаємодії між собою, а транзакціями легше керувати. Тому він добре працює для додатків з чіткими і стабільними вимогами [22, 23].

Модуль використовує бібліотеку DJL для роботи з реалізованою

моделлю обробки природної мови.

Для опису функціонального складу модуля була розроблена функціональна модель у вигляді діаграм потоків даних (DFD).

Контекстна діаграма потоків даних модуля наведено на рисунку 4.1.

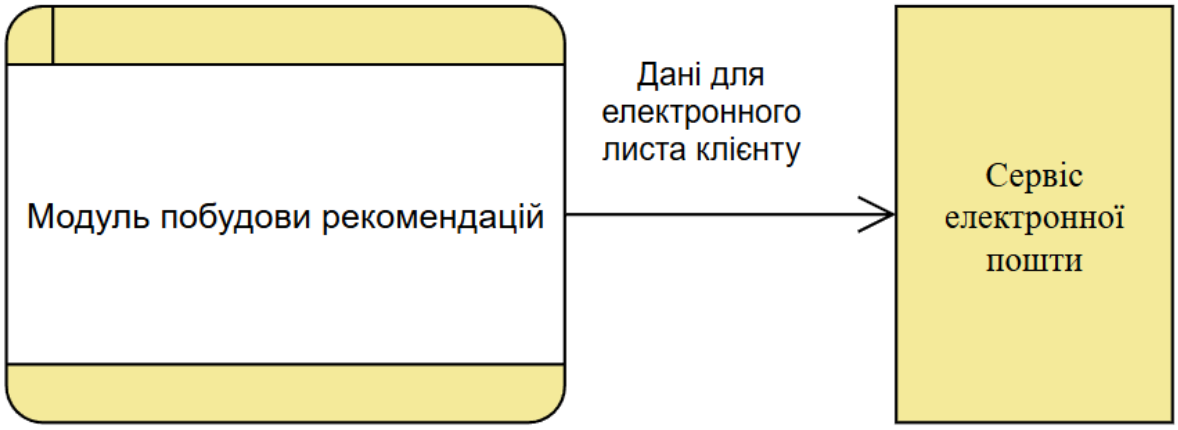


Рисунок 4.1 – Контекстна DFD модуля побудови рекомендацій

Як можна побачити з контекстної діаграми, модуль побудови рекомендацій має зовнішні зв’язки тільки з сервісом електронної пошти, тому що він викликається автоматично з періодичністю вказаною у таблиці 1.1. А генеровані персоналізовані рекомендації модуль розміщує у складі сформованого електронного листа відповідному клієнту. Всі необхідні дані для відправки електронного листа передаються сервісу електронної пошти. Таким чином клієнт взагалі не взаємодіє з модулем безпосередньо, а рекомендації щодо житла отримує їх від сервісу електронної пошти у вигляду електронного листа.

Схема функціональної структури модуля наведена на рисунку 4.2 у вигляді DFD першого рівня декомпозиції.

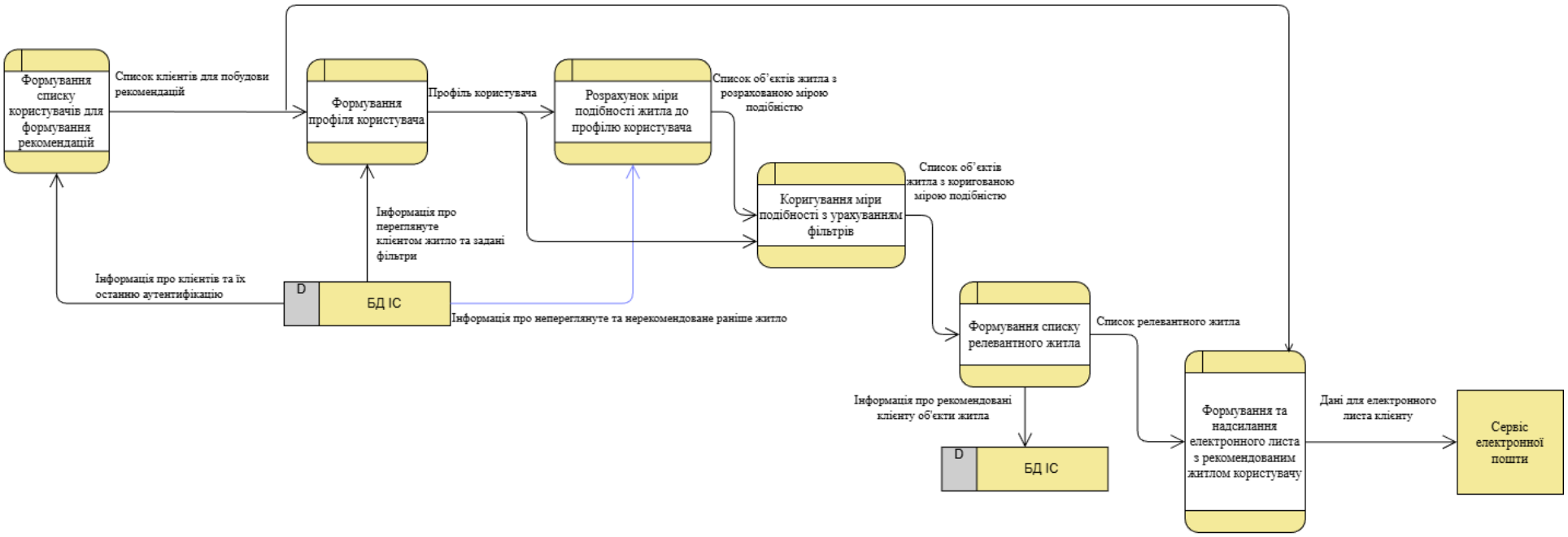


Рисунок 4.2 – Діаграма декомпозиції першого рівня контекстної DFD модуля побудови рекомендацій

Ця діаграма включає шість основних функцій модуля побудови рекомендацій:

а) функція «Формування списку користувачів для формування рекомендацій» визначає, якім користувачам треба відправити листи з рекомендованим житлом та ініціюється процес створення рекомендацій для них;

б) функція «Формування профіля користувача» формує профіль користувача на основі переглянутого житла та заданих фільтрів;

в) функція «Розрахунок міри подібності житла до профілю користувача» розраховує ступінь відповідності непереглянутого об’єкта житла профілю користувача з використанням методів косинусної подібності та BERT-подібної моделі;

г) функція «Коригування міри подібності з урахуванням фільтрів» розраховує ступінь відповідності непереглянутого об’єкта житла фільтрам. Отримана оцінка відповідності застосовується для зниження оцінки відповідності профілю в залежності від рівня порушення фільтрів користувача;

ґ) функція «Формування списку релевантного житла» реалізує формування рейтингу непереглянутих об’єктів та відбір п’яті перших об’єктів у рейтингу;

д) функція «Формування та надсилання електронного листа з рекомендованим житлом користувачу» формує дані електронного листа та його вміст у вигляді html-коду для відображення короткої інформації про рекомендовані об’єкти житла. Крім того html-код реалізує гіперпосилання на сторінки ІС для більш детального перегляду інформації про об’єкт. Дані електронного листа та вміст у вигляді html-код передаються на вхід поштового сервісу.

Модуль побудови рекомендацій, пов’язаний з іншими частинами інформаційної системи через базу даних. Так він бере інформацію від модуля облікових записів, а модуль пошуку та перегляду житла фіксує дії

користувачів для модуля рекомендацій.

Діаграма класів реалізованого модуля рекомендацій ІС агентства нерухомості представлена нижче на рисунку 4.3.

На даній діаграмі присутні такі класи, що відносяться до модуля рекомендацій ІС агентства нерухомості або пов’язані з ним:

а) Mailing – основний клас модуля, який ініціює основні його процеси від відбору користувачів, яким треба надіслати листи, до надсилання цих листів;

б) User – містить інформацію про зареєстрованого користувача: його особисті дані, останню, активність, розміщені об'єкти, історію переглядів та інше;

в) Property – містить інформацію про об'єкт нерухомості з повним набором характеристик: опис, орендна плата, розташування, кількість кімнат і ліжок, площа, поверх та інше. Кожен об'єкт пов'язаний з власником, списком користувачів які його переглядали, обладнанням та іншим;

г) Filter – містить інформацію про фільтри, які задавав користувач, пов’язаний з користувачем та обладнанням;

ґ) Profile – агрегує інформацію про вподобання користувача на основі переглянутих об’єктів житла та заданих фільтрів;

д) Equipment – містить інформацію про обладнання та додаткові параметри, пов’язаний з об’єктами житла та фільтрами;

е) BertTextToVector – перетворює текстові описи в числові вектори за допомогою BERT;

є) CosineSimilarity – на основі непереглянутого житла та профілю користувача видає список найрелевантнішого житла для користувача з використанням методів косинусної подібності та BERT-подібної моделі;

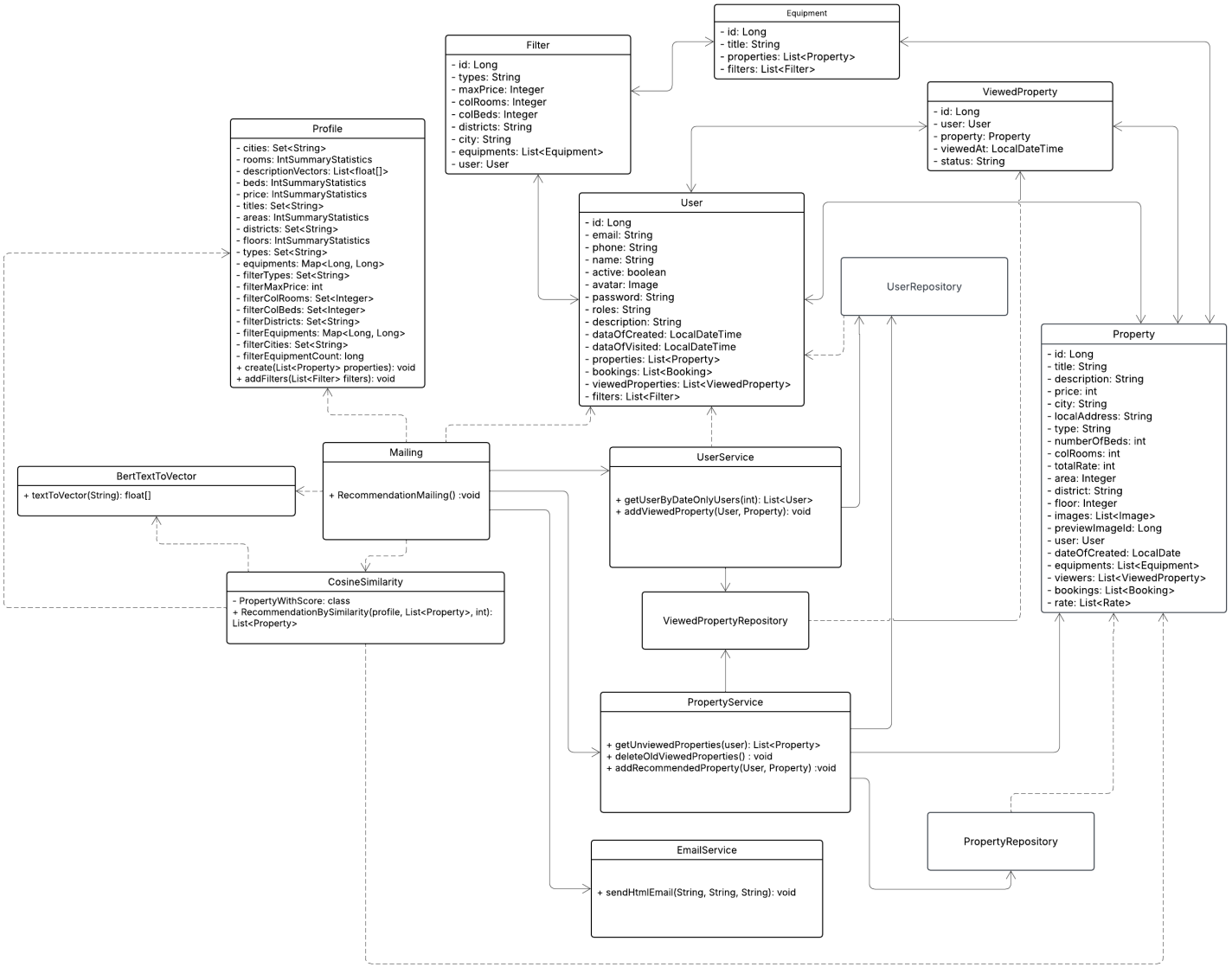


Рисунок 4.3 – Діаграма класів реалізованого модуля

ж) UserService – відповідає за роботу з користувачами. Представлені методи дозволяють модулю вибирати користувачів за датою їх останньої активності, що використовується для відправки листів відповідно до дати їх останньої активності та записувати, яке житло переглядав користувач;

з) PropertyService – управляє об'єктами нерухомості, надає модулю метод для отримання списку непереглянутих користувачем об'єктів, метод для видалення інформації про перегляди та минулі рекомендації старше п’яти місяців та метод для створення зв’язку між користувачем та об’єктом, який йому вже рекомендували;

и) EmailService – сервіс для відправи електронних листів. Використовується для надсилання листів з рекомендованим житлом;

і) ViewedProperty – містить інформацію про перегляди об’єктів житла зареєстрованими користувачами та те що житло вже було рекомендовано користувачу.

4.2 Експериментальна перевірка наукових результатів

Для перевірки коректності запропонованої ІТ було проведено експериментальне тестування в умовах, наближених до реальної експлуатації системи. До ІС агентства нерухомості були внесені п'ятнадцять тестових об'єктів житла, які представлені на рисунку 4.4. Після чого, було змодельовано поведінку користувача системи, яка включала такі дії:

а) перегляд двох об'єктів житла, інформацію про які можна побачити на рисунку 4.5;

б) застосування двох фільтрів для пошуку об'єктів житла, значення полів яких, можна побачити на рисунку 4.6.

id	city	col_rooms	description	local_address	col_beds	preview_image_id	price	type	user_id	area	district	floor	equipments
152	Харків	2	В квартирі 2 кімнати, Технологія будівництв...	просп. Ювілейний, 10	2	102	6000	Квартира	1	42	Салтівський	4	З ремонтом, Інтернет, Ліфт, Пральна машина
153	Харків	2	В квартирі 2 кімнати, Технологія будівництв...	просп. Ювілейний, 20	2	103	6000	Квартира	1	42	Салтівський	4	З ремонтом, Інтернет, Ліфт, Пральна машина
154	Харків	2	Здам в оренду 2 кімнатну квартиру на комф...	просп. Ювілейний, 51	2	104	5000	Квартира	1	43	Салтівський	1	З ремонтом, Інтернет, Ліфт, Перший поверх, Пральна машина
155	Харків	3	Довготривала оренда. Ювілейний, 59 (пору...	просп. Ювілейний, 59	3	105	5000	Квартира	1	33	Салтівський	3	З ремонтом, Інтернет, Ліфт, Пральна машина
156	Харків	1	Сдам 1 комнатную квартиру. В квартире ест...	вул. Роганська, 15	2	106	4000	Квартира	1	17	Індустріальний	5	З ремонтом, Інтернет, Ліфт, Поруч з метро
157	Харків	3	Генераторы ! Низкий этаж ! Аренда 3к Lux А...	вул. Гімназійна, 26	5	107	14000	Квартира	1	54	Основянський	5	З ремонтом, Інтернет, Парковка, Пральна машина
158	Харків	1	Здам свою кімнату в блоці з 6 кімнат. Кімнат...	вул. Роганська, 30	2	108	2000	Кімната	1	13	Індустріальний	1	Дозволено паління, Ліфт, Перший поверх, Поруч з метро
159	Харків	1	Сдам кімнату. 4 поверх п'яти поверхового б...	вул. Роганська, 35	2	109	1700	Кімната	1	10	Індустріальний	4	Дозволено паління, Ліфт, Поруч з метро
160	Харків	1	Сдам кімнату. 4 поверх п'яти поверхового б...	вул. Роганська, 40	2	110	1700	Кімната	1	10	Індустріальний	4	Дозволено паління, Ліфт, Поруч з метро
161	Харків	1	Сдам 1 комнату в частном секторе. 8 кв.м. Н...	вул. Роганська, 43	1	111	3000	Кімната	1	8	Індустріальний	1	Дозволено паління, Ліфт, Перший поверх, Поруч з метро
162	Харків	1	Здаються затишні гостінки (номери), розташ...	вул. Гімназійна, 39	2	112	15000	Кімната	1	70	Основянський	1	Дозволено паління, Ліфт, Перший поверх, Поруч з метро, Пральна машина
163	Харків	4	сдам дом со всеми удобствами!стиралка пыл...	вул. Гімназійна, 43	10	113	18000	Будинок	1	100	Основянський	NULL	Можна з деякими тваринами, Можна з дітьми, Парковка, Пральна машина
164	Харків	4	Сдам дом, 2х этажный, 2 х комнатный , с об...	вул. Гімназійна, 11	5	114	16800	Будинок	1	58	Основянський	NULL	Можна з деякими тваринами, Можна з дітьми, Парковка, Пральна машина
165	Київ	5	Сдается в долгосрочную аренду домик (1ко...	вул. Гімназійна, 2	5	115	17000	Будинок	1	55	Основянський	NULL	Можна з деякими тваринами, Можна з дітьми, Парковка, Пральна машина
166	Київ	5	Здається в довгострокову оренду будиночо...	вул. Гімназійна, 23	5	116	17000	Будинок	1	55	Основянський	NULL	Можна з деякими тваринами, Можна з дітьми, Парковка, Пральна машина

Рисунок 4.4 – Інформація про об’єкти житла, які були використані для експерименту

	id	viewed_at	property_id	user_id	status
▶	1	2025-12-08 19:17:45.576054	152	1	viewed
	2	2025-12-08 19:18:00.089041	154	1	viewed

Рисунок 4.5 – Номери відвіданих об’єктів житла

	filter_id	city	col_beds	col_rooms	districts	max_price	types	user_id	equipments
▶	1	Харків	2	2	Салтівський	6300	Квартира	1	Інтернет, Ліфт
	2	Харків	2	3	Салтівський	7000	Квартира	1	Інтернет, Перший поверх

Рисунок 4.6 – Значення полів фільтрів, заданих при пошуку

Виходячи з цих даних, метод буде працювати вірно, якщо сформує рекомендаційний лист з об’єктами 153, 155, 156 та 157, бо вони є найбільш схожими на переглянуті об’єкти та більш відповідають фільтрам ніж інші непереглянуті об’єкти житла.

Після додавання даних був запущений механізм формування рекомендаційного листа та після його отримання можна буде побачити, чи попали необхідні об’єкти житла до списку рекомендованих об’єктів. Сформований модулем html-код змісту рекомендаційного листа можна побачити у лістингу А.1 додатку А.

При формуванні коду використовувалися дані об’єктів житла, відібраних модулем для рекомендації, та ім’я клієнта, якому надсилається електронний лист. Відображення цього коду можна побачити на скріншоті рекомендаційного листа, представленого на аркушах рисунку 4.7.

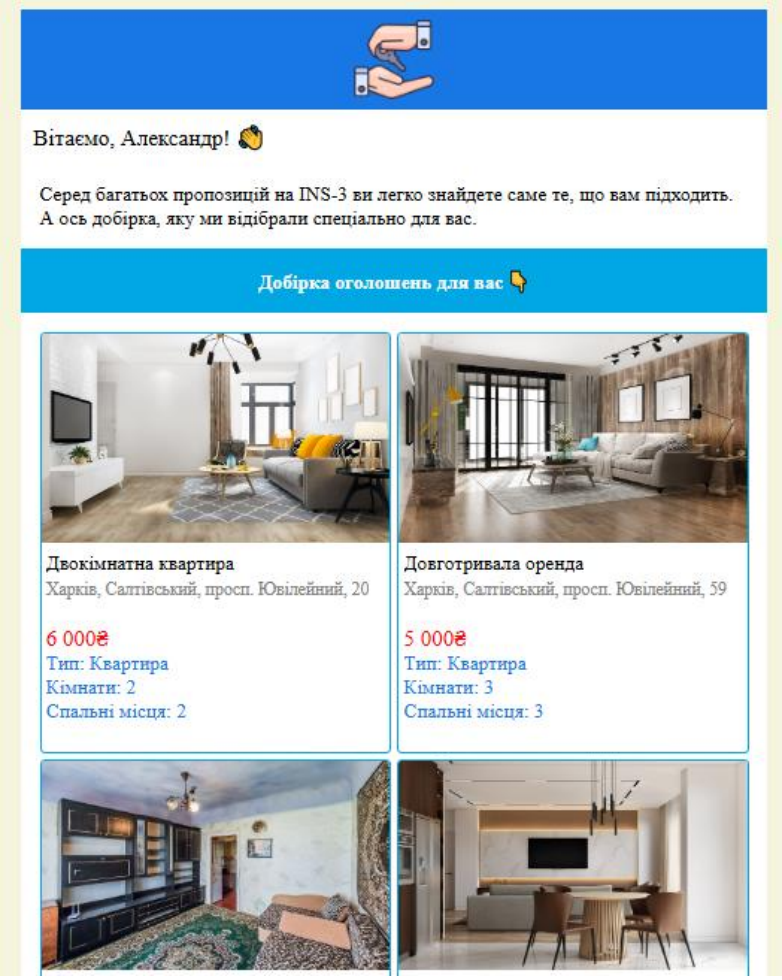


Рисунок 4.7 – Скріншот рекомендаційного листа

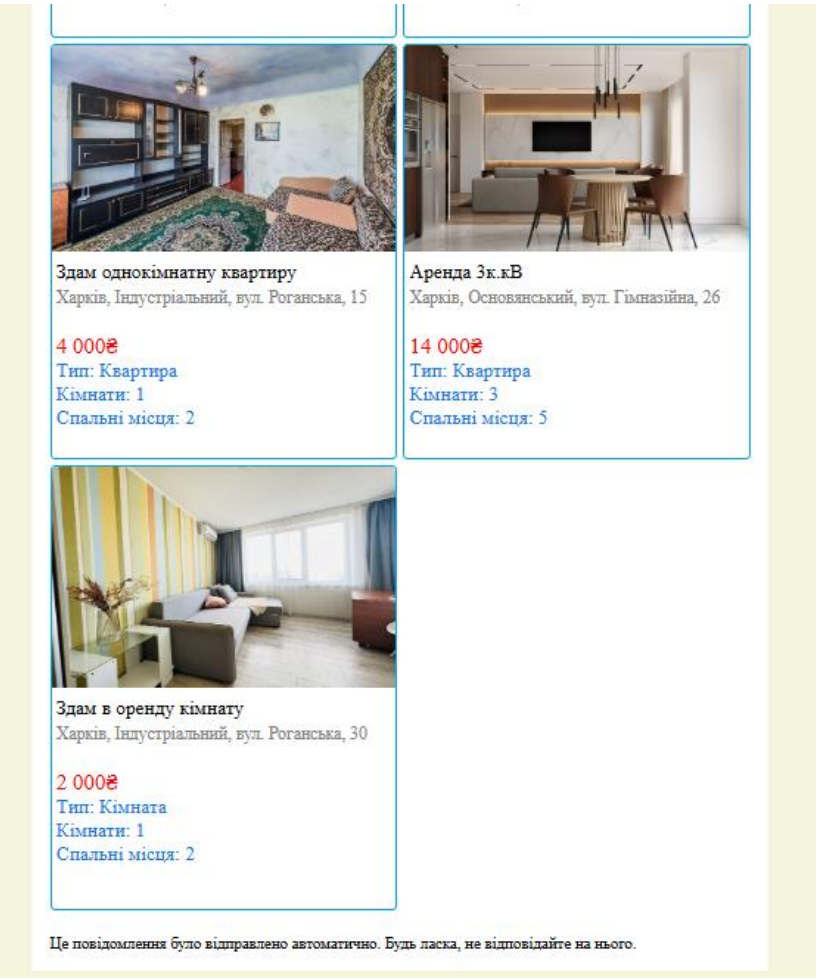


Рисунок 4.7, аркуш 2

Аналіз отриманих результатів свідчить про повну відповідність фактичного виводу системи очікуваним результатам. Усі чотири прогнозовані об'єкти (153, 155, 156, 157) зайняли перші чотири позиції у рекомендаційному листу, що підтверджує коректність функціонування розробленого модуля рекомендацій.

Результати експериментальної перевірки демонструють належну роботу модуля рекомендацій у складі ІС агентства нерухомості. Модуль побудови рекомендаций успішно ідентифікує найбільш релевантні об'єкти житла на основі аналізу переглянутих користувачем об’єктів житла та параметрів, заданих ним, фільтрів, що підтверджує ефективність обраних методів та розробленої інформаційної технології.

ВИСНОВКИ

У межах кваліфікаційної роботи було проведено дослідження методів і моделей, придатних для використання під час побудови рекомендацій клієнтам в ІС агентства нерухомості.

У роботі сформульовано проблему побудови рекомендацій клієнтам щодо вибору нерухомості, виконано аналіз існуючих методів та моделей, які необхідні для побудови рекомендацій.

Під час виконання роботи було проведене експериментальне дослідження трьох методів розрахунку ступеня подібності об'єктів та двох методів і моделі перетворення текстових атрибутів об’єктів на вектори з метою визначення кращих для використання при побудові рекомендацій щодо житла у ІС агентства нерухомості. Розроблено системи критеріїв вибору методів розрахунку ступеня подібності об’єктів та методів для перетворення текстових атрибутів об’єктів на вектори.

Найкращі результати показав метод косинусної подібності та BERT-подібна модель sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2.
З використанням цих методів була розроблена ІТ побудови рекомендацій клієнтам в ІС агентства нерухомості та виконана програмна реалізація ІТ у вигляді модуля побудови рекомендацій.

Також проведена експериментальна перевірка отриманих наукових результатів, яка підтвердила ефективність обраних методів та розробленої ІТ.

Результати проведеного дослідження можна використовувати при реалізації рекомендаційних функцій в ІС, які використовуються у сфері нерухомості.

Доцільним вважаю продовження досліджень з метою визначення ефективних методів для інформаційних технологій формування персоналізованих рекомендацій, які мають бути реалізовані безпосередньо на сторінках ІС , без обмеження лише форматом рекомендаційних листів.

Результати роботи були опубліковані на п’ятій міжнародній науково-теоретичній конференції «Current scientific goals, approaches and challenges» у форматі тез доповіді на тему «Дослідження методів побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості» [24].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В., Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Інформаційні управляючі системи та технології»). Харків: ХНУРЕ, 2021. 24 с.

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 22.06.2015. Київ: ДП «УкрНДНЦ», 2016. 26 с.

3. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. Вид. офіц. Київ : УкрНДНЦ, 2016. 16 с.

4. Kim J., Choi I., Li Q. Customer Satisfaction of Recommender System: Examining Accuracy and Diversity in Several Types of Recommendation Approaches. Sustainability. 2021. Т. 13, № 11. С. 6165. URL: https://doi.org/10.3390/su13116165 (дата звернення: 10.12.2025).

5. Deep Learning Based Recommender System / S. Zhang та ін. ACM Computing Surveys. 2019. Т. 52, № 1. С. 1–38. URL: https://doi.org/10.1145/3285029 (дата звернення: 01.11.2025).

6. Дослідження методів побудови рекомендаційних систем для розв'язання задачі вибору найбільш релевантного відео при створенні віртуальних арт-композицій / A. Kuliahin та ін. Системи управління, навігації та зв'язку. Збірник наукових праць : матеріали Міжнар. наук. конф., м. м. Полтава, 29 листоп. 2022 р. м. Полтава, 2022. С. 94–99. URL: https://doi.org/10.26906/sunz.2022.4.094 (дата звернення: 13.11.2025)..

7. Ryngksai I., Chameikho L. Recommender Systems: Types of Filtering Techniques. International Journal of Engineering Research & Technology (IJERT).

2014. Т. 3, № 11. С. 251–254. URL: https://www.ijert.org/research/recommender-systems-types-of-filtering-techniques-IJERTV3IS110197.pdf (дата звернення: 11.12.2025).

8. Çano E., Morisio M. Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis. 2017. Т. 21, № 6. С. 1487–1524. URL: https://doi.org/10.3233/ida-163209 (дата звернення: 01.11.2025).

9. Gómez J., Vázquez P.-P. An Empirical Evaluation of Document Embeddings and Similarity Metrics for Scientific Articles. Applied Sciences. 2022. Т. 12, № 11. С. 5664. URL: https://doi.org/10.3390/app12115664 (дата звернення: 12.11.2025).

10. Levy A., Shalom B. R., Chalamish M. A guide to similarity measures and their data science applications. Journal of Big Data. 2025. Т. 12, № 1. URL: https://doi.org/10.1186/s40537-025-01227-1 (дата звернення: 12.11.2025).

11. Çolakoğlu H. B. On the distance formulae in the generalized taxicab geometry. Turkish journal of mathematics. 2019. Т. 43, № 3. С. 1578–1594. URL: https://doi.org/10.3906/mat-1809-78 (дата звернення: 12.11.2025).

12. Etherington T. R. Mahalanobis distances for ecological niche modelling and outlier detection: implications of sample size, error, and bias for selecting and parameterising a multivariate location and scatter method. PeerJ. 2021. Т. 9. С. e11436. URL: https://doi.org/10.7717/peerj.11436 (дата звернення: 11.12.2025).

13. Aizawa A. An information-theoretic perspective of tf–idf measures. Information Processing & Management. 2003. Т. 39, № 1. С. 45–65. URL: https://doi.org/10.1016/s0306-4573(02)00021-3 (дата звернення: 11.12.2025).

14. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) : тези доп., м. Міннеаполіс, черв. 2019 р. Minneapolis : Association for Computational Linguistics, 2019. С. 4171–4186. URL: https://aclanthology.org/N19-1423/ (дата звернення: 29.11.2025).

15. Tian M., Ekstrand M. D. Estimating Error and Bias in Offline Evaluation Results. CHIIR '20: Proceedings of the 2020 Conference on Human Information Interaction and Retrieval, м. Vancouver, 14 берез. 2020 р. New York, 2020. С. 392–396. URL: https://dl.acm.org/doi/10.1145/3343413.3378004 (дата звернення: 24.11.2025).

16. Comparative Study of Filtering Methods for Scientific Research Article Recommendations / D. El Alaoui та ін. Big Data and Cognitive Computing. 2024. Т. 8, № 12. С. 190. URL: https://doi.org/10.3390/bdcc8120190 (дата звернення: 11.12.2025).

17. Gebremeskel G. G., de Vries A. P. Recommender Systems Evaluations: Offline, Online, Time and A/A Test. Conference and Labs of the Evaluation Forum : матеріали Міжнар. наук. конф., м. Évora, 5 верес. 2016 р. Aachen, 2016. С. 642–656. URL: https://ceur-ws.org/Vol-1609/16090642.pdf (дата звернення: 14.11.2025).

18. A/B testing: A systematic literature review / F. Quin та ін. Journal of Systems and Software. 2024. С. 112011. URL: https://doi.org/10.1016/j.jss.2024.112011 (дата звернення: 11.12.2025).

19. Sohangir S., Wang D. Improved sqrt-cosine similarity measurement. Journal of Big Data. 2017. Т. 4, № 1. URL: https://doi.org/10.1186/s40537-017-0083-6 (дата звернення: 11.12.2025).

20. Mussabayev R. Optimizing Euclidean Distance Computation. Mathematics. 2024. Т. 12, № 23. С. 3787. URL: https://doi.org/10.3390/math12233787 (дата звернення: 11.12.2025).

21. A modified and weighted Gower distance-based clustering analysis for mixed type data: a simulation and empirical analyses / P. Liu та ін. BMC Medical Research Methodology. 2024. Т. 24, № 1. URL: https://doi.org/10.1186/s12874-024-02427-8 (дата звернення: 11.12.2025).

22. Mooghala S. A Comprehensive Study of the Transition from Monolithic to Micro services-Based Software Architectures. Journal of Technology and Systems. 2023. Т. 5, № 2. С. 27–40. URL: https://doi.org/10.47941/jts.1538 (дата

звернення: 09.12.2025).

23. Kambhammettu A. P. Monolithic versus Microservice Architectures: A Comparative Analysis for Enterprise Applications. European Journal of Computer Science and Information Technology. 2025. Т. 13, № 51. С. 65–75. URL: https://doi.org/10.37745/ejcsit.2013/vol13n516575 (дата звернення: 11.12.2025).

24. Требунських О., Борисенко Т. Дослідження методів побудови рекомендацій клієнтам в інформаційній системі агентства нерухомості. V International Scientific and Theoretical Conference «Currentscientific goals, approaches and challenges» : матеріали Міжнар. наук. конф., м. Dresden, 12 груд. 2025 р. Dresden, 2025. С. 296–299. URL: https://previous.scientia.report/index.php/archive/issue/view/12.12.2025 (дата звернення: 12.12.2025).